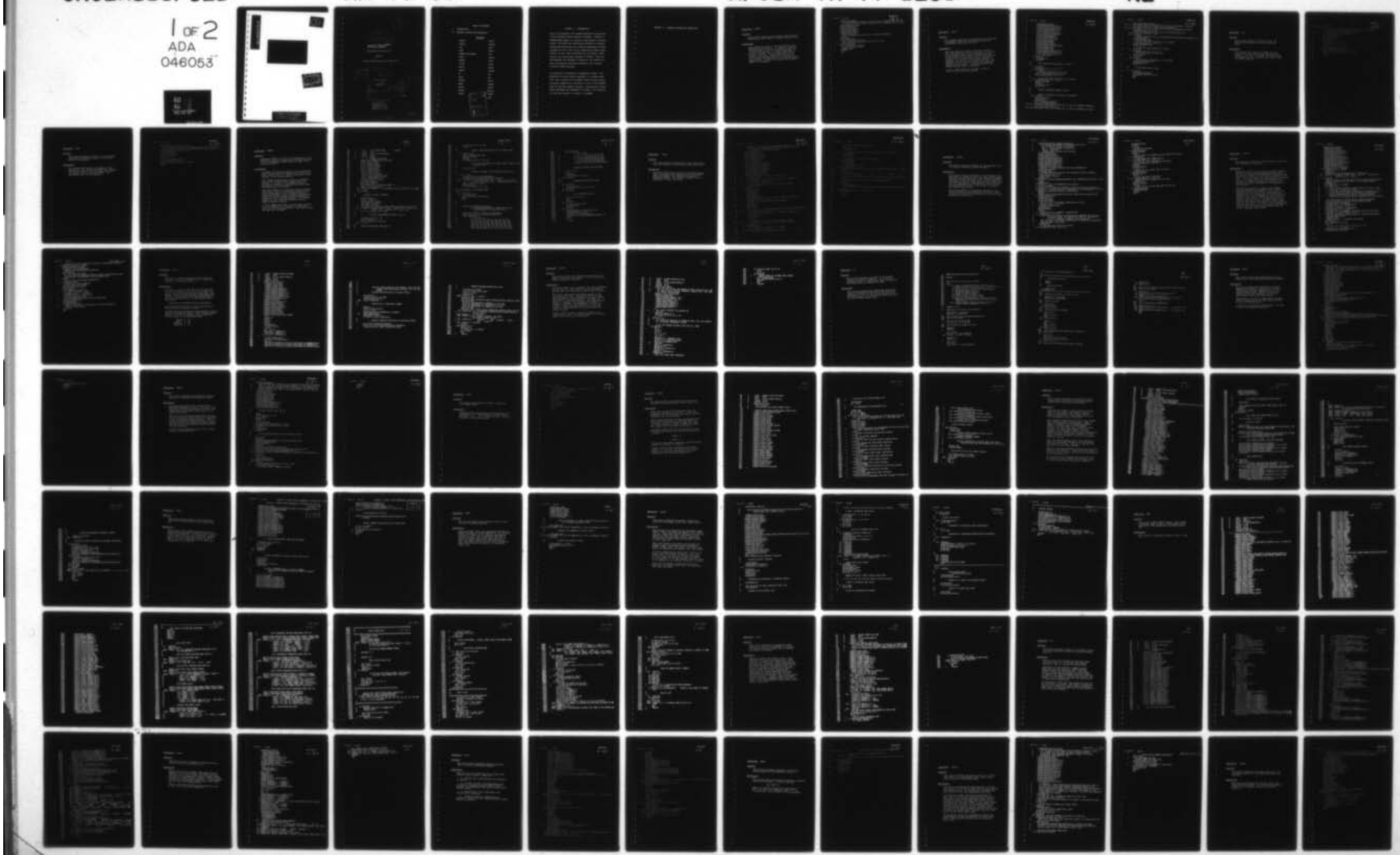
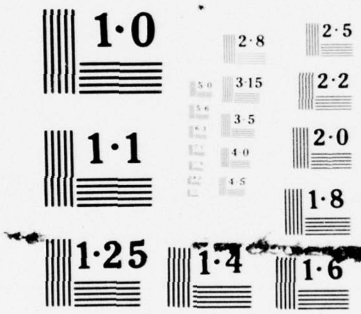
AD-A046 053   DECISION SYSTEMS   DAYTON OH   F/G 15/5
THE INVENTORY SYSTEM SIMULATOR (INSSIM). VOLUME II PROGRAM LIST--ETC(U)
AUG 77   W S DEMMY   F49620-77-C-0063
UNCLASSIFIED   RM-77-03   AFOSR-TR-77-1235   NL

1 OF 2
ADA
046053




1·0
2·8
2·5
3·15
2·2
5·0
5·6
3·5
2·0
4·0
1·1
4·5
1·8
1·25
1·4
1·6

3

18 AFOSR

19 TR-77-1235

9 Technical rept.,

Approved for public release;
distribution unlimited.

6

THE INVENTORY SYSTEM SIMULATOR

(INSSIM).

VOLUME II: PROGRAM LISTINGS AND NARRATIVES.

15 F49620-77-C-0063

16 2304

by

17 A5

10

W. Steven Demmy

11

August 1977

12 119p.

14 RM-77-93

Decision Systems
3575 Charlene Drive
Dayton, Ohio 45432

1473

393 082

TABLE OF CONTENTS

ROUTINES

B

CHAPTER I. INTRODUCTION

This is the second of two volumes describing the features of the Inventory System Simulator (INSSIM). INSSIM is a FORTRAN-based model of a general single-echelon inventory system. The model was specifically designed to evaluate alternative forecasting and inventory management policies proposed for use in the EOQ Buy Computation System (D062). Volume I of this report describes the philosophy, organization, and input/output features of INSSIM. This volume documents the statement listings of each INSSIM program, and provides narratives describing the function of each of these routines.

The routines are presented in alphabetical order. Documentation for each routine consists of a program narrative, and a listing of the FORTRAN code for that routine. Additional comments are included as a part of the FORTRAN code for the more complex routines. Definitions of major INSSIM variables are documented in Volume I, and familiarity with the contents of Volume I is assumed.

CHAPTER II. PROGRAM LISTINGS AND NARRATIVES

Subroutine: BUDGET

Function:

This routine updates period counters and obligation accumulators, and creates two budget review events.

Description:

This routine is called at the conclusion of each budget revision interval. It zeros the accumulator IOBLIG used to record obligations within the current budget period. It then creates type 6 and type 7 events, and puts these events on the Future Events List. These events are used to revise buy guidelines within the budget period, and to signal the beginning of a new budget period, respectively. Finally, the budget period counter, NBPER, is incremented.

1   12-08-76   20.399

BUDGET
p 1 of 1

```
        SUBROUTINE BUDGET
C         AT THE CONCLUSION OF EACH BUDGET REVISION INTERVAL,THIS ROUTINE
C         IS CALLED TO UPDATE PERIOD COUNTERS AND OBLIGATION ACCUMULATORS
        COMMON/IOBLIG/IOBLIG
        COMMON/ITHQ/ITHQ
        COMMON/IDTHQ/IDTHQ
        COMMON/NBPER/NBPER
        COMMON/ITDIV/ITDIV
        COMMON/ITIME/ITIME
C         RESET COUNTERS AND CALL GUIDE TO RESET GUIDELINES
        IOBLIG=0
        ITDIV=ITIME+10
        CALL ENTER(ITDIV,7,0,0,0)
C         PLACE BUDGET REVIEW TRANSACTION ON FEL LIST
     12 ITHQ=ITHQ+IDTHQ
        CALL ENTER(ITHQ,8,0,0,0)
C         UPDATE PERIOD COUNTER
        NBPER=NBPER+1
        RETURN
        END
```

Subroutine: CANCLB

Function:

This routine simulates the cancellation of backorders by customers which have been forced to wait an unacceptably long period of time.

Description:

When this routine is called, the stock status of each item being simulated is reviewed. If any item has a backorder that is older than MBODAYS days, routine CANCLB takes action to reflect cancellation of the associated requisition by the requesting customer. To do this, data elements defining the backorder are removed from the Backorder File, and Subroutine CUM is called to update the associated performance statistics. Statistics affected are: IRATON, ICANCL, IBACKT, and IBACKI.

Finally, CANCLB creates a new type 3 event to occur at time ITCANB plus IDCANB.

CANCLB
p. 1 of 2

```
01  12-08-76    19.916
         SUBROUTINE CANCLB
         COMMON/IBACPT/IBACPT(200)
         COMMON/ICANCL/ICANCL(30,3)
         COMMON/IPRIOR/IPRIOR(200)
         COMMON/IQTYB/IQTYB(200)
         COMMON/ITMBAC/ITMBAC(200)
         COMMON/ILOCBK/ILOCBK(200)
         COMMON/NLOCBK/NLOCBK
         COMMON/NBOPT/NBOPT(50)
         COMMON/NBOTU/NBOTU(50)
         COMMON/ITIME/ITIME
         COMMON/ITDAY/ITDAY
         COMMON/MBODAY/MBODAY
         COMMON/ITCANB/ITCANB,IDCANB
         COMMON/IDBUG/IDBUG
         COMMON/NITEM/NITEM
         COMMON/IBACKT/IBACKT(30,3)
         COMMON/IBACKI/IBACKI(30,3)
         COMMON/IRATON/IRATON(30,3)
         DO 500 N=1,NITEM
      50 CONTINUE
         IPT=NBOPT(N)
         JPT=IPT
  C
  C         ARE THERE ANY BACKORDERS ON ITEM N
  C
     100 CONTINUE
         IF(NBOPT(N).LE.0) GO TO 500
     110 CONTINUE
  C         COMPUTE BACKORDER AGE IN DAYS
         IAGE=(ITIME-ITMBAC(JPT))/ITDAY
         IF(IAGE.GT.MBODAY) GO TO 150
  C
  C         IS JPT THE LAST BACKORDER ON THE CHAIN
     140 IF(IBACPT(JPT).LE.0) GO TO 500
  C      UPDATE INDICES
         IPT=JPT
         JPT=IBACPT(JPT)
         GO TO 110
  C
  C         CANCEL BACKORDER LOCATED AT JPT
  C
  C
  C         UPDATE STATISTICS TO REFLECT CANCELATION
     150 CONTINUE
         IQTY=IQTYB(JPT)
         JQTY=-IQTY
         NBOTU(N)=NBOTU(N)-IQTY
         CALL CUM(IRATON,IQTY,N)
1457 DO LOOP INDEX N MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
         CALL CUM(ICANCL,IQTY,N)
1457 DO LOOP INDEX N MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
```

CANCLB

p 2 of 2

```
01  12-08-76    19.916

          CALL CUM(IBACKT,JQTY,N)
 1457 DO LOOP INDEX N MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
          IF(IPRIOR(JPT).EQ.1) CALL CUM(IBACKI,JQTY,N)
 1457 DO LOOP INDEX N MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
          IF(IDBUG.NE.1) GO TO 160
          WRITE(6,8000)ITIME,N,IQTY,IPRIOR(JPT),ITMBAC(JPT),JPT
     8000 FORMAT(7H CANCEL,I10,5I8)
      160 CONTINUE
    C         RECORD LOCATION JPT AS AVAILABLE FOR FILING OTHER BACKORDER DAT
          NLOCBK=NLOCBK+1
          ILOCBK(NLOCBK)=JPT
    C         IS JPT THE FIRST BACKORDER ON THE CHAIN
          IF(JPT.NE.NBOPT(N)) GO TO 170
          NBOPT(N)=IBACPT(JPT)
          GO TO 50
    C         IS JPT THE LAST ENTRY ON THE CHAIN
      170 CONTINUE
          IF(IBACPT(JPT).NE.0) GO TO 180
    C         SET IPT AS LAST ON THE CHAIN
          IBACPT(IPT)=0
          GO TO 500
      180 CONTINUE
    C         THEN JPT IS IN THE MIDDLE OF THE CHAIN
          IBACPT(IPT)=IBACPT(JPT)
          JPT=IBACPT(IPT)
          GO TO 110
    C
    C         CREATE NEXT REVIEW EVENT
    C
      500 CONTINUE
          ITCANB=ITCANB+IDCANB
          CALL ENTER(ITCANB,3,0,0,0)
          RETURN
          END
```

Subroutine: CUM

Function:

This routine updates the action, units, and dollar counts associated with the statistic ISTAT.

Description:

The variable KFSN defines the number of the items involved and IQTY defines the associated quantity. If IQTY is negative, the action count is decreased by one. The units and dollar counts are always increased.

1   02-11-77   10.619

CUM
p.1 of 1

```
      SUBROUTINE CUM(ISTAT,IQTY,KFSN)
C       THIS ROUTINE UPDATES THE ITEMS,UNITS,AND DOLLAR COUNTS
C       ASSOCIATED WITH THE STATISTIC ISTAT
      DIMENSION ISTAT(30,3),UCOST(50)
      COMMON/ITINV/ITINV
      COMMON/UCOST/UCOST
      COST=UCOST(KFSN)
      I=ITINV
      IF(IQTY.LT.0) GO TO 21
      ISTAT(I,1)=ISTAT(I,1)+1
      GO TO 22
   21 ISTAT(I,1)=ISTAT(I,1)-1
   22 ISTAT(I,2)=ISTAT(I,2)+IQTY
      DOLLAR=COST*FLOAT(IQTY)
      ISTAT(I,3)=ISTAT(I,3)+IFIX(DOLLAR)
      RETURN
      END
```

Subroutine: CUMB

Function:

This routine updates the units and dollar counts associated with statistic ISTAT. The actions count is not affected.

Description:

The variable KFSN defines the number of items involved and IQTY defines the associated quantity. When called, Subroutine CUMB updates the units and dollar counts of statistic ISTAT. The actions/FSN count is not affected.

CUMB,
p. 1 of 1

```
02-11-77    10.698

      SUBROUTINE CUMB(ISTAT,IQTY,KFSN)
C       THIS ROUTINE UPDATES THE UNITS AND DOLLAR COUNTS ASSOCIATED
C       WITH STATISTIC ISTAT. THE ACTIONS COUNT IS NOT EFFECTED
      DIMENSION ISTAT(50,3),UCOST(50)
      COMMON/ITINV/ITINV
      COMMON/UCOST/UCOST
      COST=UCOST(KFSN)
      I=ITINV
      ISTAT(I,2)=ISTAT(I,2)+IQTY
      DOLLAR=COST*FLOAT(IQTY)
      ISTAT(I,3)=ISTAT(I,3)+IFIX(DOLLAR)
      RETURN
      END
```

Subroutine: DEMPAR

Function:

Subroutine DEMPAR is called at the beginning of each simulation quarter to generate the specific requisition and serviceable return events to occur in the current quarters.

Description:

In DEMPAR, the variable IDPER is first incremented. This variable defines the historical period in the Demand Driver File to be used in generating demand activity in the current quarter.

Next, DEMPAR generates type 4 event to represent the arrival of serviceable returns within the quarter. At present, it is assumed that all serviceable returns are generated one month after the beginning of the quarter.

Subroutine DEMPAR next generates specific requisition events to occur in the current quarter, and places these events on the Future Events List. Subroutine GETREQ is used to generate requisition sizes with the same probability distributions as observed in AFLC inventory systems. See Volume I, Chapter VI, for a detailed discussion of the demand generation process.

Finally, DEMPAR generates a type 12 event to occur at a time IDPER in the future, and places this event on the Future Events List. This action causes the next call to DEMPAR.

1 of 3

```
$       IDENT   WP1279,XRS/DEMMY          DEMPAR
$       FORTY   DECK
$       PRMFL   C*,W,S,REQS/DEMPAR.O
$       REMOTE  P*,AC
$       REMOTE  $$,AC
C               DEMPAR FOR DODSIM
      SUBROUTINE DEMPAR(IDPER,IP4,IP5)
      COMMON/IGBUG/IGBUG
      COMMON/ITIME/ITIME
      COMMON/ITWEEK/ITWEEK
      COMMON/ITMNTH/ITMNTH
      COMMON/ITQTR/ITQTR
      COMMON/ITYEAR/ITYEAR
      COMMON/NITEM/NITEM
       COMMON/NDENT/NDENT(1)
       COMMON/IDEMND/IDEMND(1,24)
       COMMON/IRETUR/IRETUR(1,24)
       COMMON/IREQ/IREQ(1,24)
      IF(IGBUG.NE.1) GO TO 25
      IY=ITIME/ITYEAR +1
      IQT=ITIME/ITQTR +1
      ITW=(ITIME-(IY-1)*ITYEAR)/ITWEEK +1
      WRITE(6,103)IY,IQT,ITW
  103 FORMAT(12H DEMPAR-----,5X,8HYEAR NO.,I3,5X,7HQTR NO.,I3,5
     1 NO.,I3)
   25 CONTINUE
C
C         INCREMENT PERIOD COUNTERS
C
      IDPER=IDPER+1
      DO 100 N=1,NITEM
       IQTY=IDEMND(N,IDPER)
       IRET=IRETUR(N,IDPER)
       IRQ=IREQ(N,IDPER)
       IF(IGBUG.EQ.1)WRITE(6,8013)N,NDENT(N),IDPER,IQTY,IRET,IRQ
 8013  FORMAT(4X,'****DEMPAR--N=',I5,' NDENT=',I5,' IDPER=',I5,
&               ' IDEMND=',I5,' IRETRN=',I5,' IREQ=',I5)
      IF(NDENT(N).LE.0) GO TO 100
C
C                 GENERATE SERVICEABLE RETURNS, IF ANY
C
       IF(IRET.LE.0)GO TO 30
       IT=ITIME+ITMNTH
       CALL ENTER(IT,4,N,IRET,0)
 30    CONTINUE
C
C      REFLECT REQUISTION FOR ITEM N
C
```

```
            IF(IQTY.LE.Ø) GO TO 1ØØ
            ITOT=Ø
      C
      C             OBTAIN A REQUISITION SIZE IR BY MONTE CARLO
      C
         5Ø R=RANDU(.2)
            CALL GETREQ(R,IQTY,IRQ)
            ITEMP=ITOT+IRQ
            IR=IRQ
            IF(ITEMP.LE.IQTY) GO TO 6Ø
      C
      C             LIMIT LAST REQUISITION SO THAT TOTAL DEMAND IN GE
      C                EQUALS IQTY.
      C
            IR=IQTY-ITOT
      C
      C             DETERMINE ARRIVAL TIME FOR THIS REQUISITION
      C
         6Ø CONTINUE
            IT=ITIME+RANDU(.2)*FLOAT(ITQTR)
            IF(IGBUG.EQ.1)WRITE(6,8Ø23)IT,N,R,IRQ,IR
       8Ø23 FORMAT(4X,'****DEMPAR--IT=',I1Ø,'  N=',I4,'  R=',F5.3,
           &         INITAL REQ-SIZE=',I4,'  FINAL REQ-SIZE=',I4)
            CALL ENTER(IT,1,N,IR,1)
            ITOT=ITOT+IR
            IF(ITOT.LT.IQTY) GO TO 5Ø
      C
      C********CREATE NEXT DEMPAR EVENT
      C
        1ØØ CONTINUE
            IT=ITIME+ITQTR
           CALL ENTER(IT,12,IDPER,Ø,Ø)
           RETURN
           END
      C
      C
      C
            SUBROUTINE GETREQ(R,IQTY,IR)
      C             THIS ROUTINE DETERMINES A REQUISITION SIZE
      C                CORRESPONDING TO A CUM PROB. OF R.
      C
            DIMENSION IUCAT(8),CPROB(1Ø,8),IRQ(1Ø,8)
           DATA IUCAT/1,3,1Ø,31,1ØØ,316,1ØØØ,3162/
           DATA CPROB/1Ø*1.,
           &        .46,.89,8*1.Ø,
           &        .27,.52,.65,.78,.86,.91,.92,.95,.96,1.Ø,
           &        .23,.53,.7Ø,.78,.82,.9Ø,.92,.97,.99,1.ØØ,
           &        .Ø6,.38,.6Ø,.73,.81,.85,.89,.93,.98,1.ØØ,
           &        .Ø15,.23,.42,.56,.65,.7Ø,.76,.92,.97,1.ØØ,
           &        .Ø12,.17,.31,.37,.48,.55,.6Ø,.78,.87,1.ØØ,
           &        .ØØ1,.Ø3,.32,.5Ø,.6Ø,.66,.86,.96,1.ØØ,1.ØØ/
```

3 of 3

```
      DATA IRQ/1Ø*1,
     &         1, 2, 3, 7*3,
     &         1, 2, 3, 4, 5, 6, 7, 8, 9, 1Ø,
     &         1, 4, 6, 8,1Ø,13,15,2Ø,25, 3Ø,
     &         1, 5,1Ø,15,2Ø,25,3Ø,4Ø,7Ø,1ØØ,
     &         1, 5,1Ø,15,2Ø,25,3Ø,8Ø,15Ø,3ØØ,
     &         1, 5,1Ø,15,2Ø,25,3Ø,8Ø,2ØØ,6ØØ,
     &         1,2Ø,4Ø,6Ø,8Ø,1ØØ,2ØØ,3ØØ,6ØØ,7ØØ/
C
C
C               ESTABLISH DEMAND RATE CATEGORY
C
      DO 1Ø I=1,8
      IM=I
      M=IUCAT(I)
      IF(IQTY.LE.M) GO TO 3Ø
   1Ø CONTINUE
C
C               DETERMINE REQUISITION SIZE
C
   3Ø CONTINUE
      DO 4Ø I=1,1Ø
      IF(CPROB(I,IM).GE.R) GO TO 6Ø
   4Ø CONTINUE
   6Ø CONTINUE
C
C               SET REQUISITION SIZE
C
      IF(I.GT.1)GO TO 8Ø
      IR=1
      RETURN
   8Ø CONTINUE
      IF(I.LT.1Ø)GO TO 1ØØ
      IR=IRQ(1Ø,IM)
      RETURN
  1ØØ CONTINUE
      DPROB=CPROB(I,IM)-CPROB(I-1,IM)
      RPROB=R-CPROB(I-1,IM)
      DRQ=IRQ(I,IM)-IRQ(I-1,IM)
      IR=IRQ(I-1,IM)+IFIX(RPROB*DRQ/DPROB+.5)
      RETURN
      END
     $ENDJOB
```

Subroutine: ENTER

Function:

This routine enters transactions on the Future Events List, and updates the associated pointer variables.

Description:

Subroutine ENTER places events on the Future Events List, and updates the associated pointer variables. These transactions are subsequently removed by subroutine REMOVE. See Volume I, Chapter IV for a discussion of these operations.

ENTER,
p. 1 of 2

```
0   02-11-77    13.706

      SUBROUTINE ENTER(KTIME,KTYPE,KFSN,KQTY,KPRIOR)
C          THIS ROUTINE ENTERS EVENTS ON THE FUTURE EVENTS LIST
C          AND UPDATES THE CHAIN STRUCTURE
      COMMON/IDBUG/IDBUG
      COMMON/ITIME/ITIME
      COMMON/NENTRY/NENTRY
      COMMON/NFEMAX/NFEMAX
      COMMON/NFIRST/NFIRST
      COMMON/NLOC/NLOC
      COMMON/NTIME/NTIME
      COMMON/NORDPT/NORDPT(50)
      COMMON/ILOCFE/ILOCFE(500)
      COMMON/JFSN/JFSN(500)
      COMMON/JPOINT/JPOINT(500)
      COMMON/JPRIOR/JPRIOR(500)
      COMMON/JQTY/JQTY(500)
      COMMON/JTIME/JTIME(500)
      COMMON/JTYPE/JTYPE(500)
C         BEGIN BY UPDATING COUNT OF ENTRIES IN FEL
      NENTRY=NENTRY+1
C         IF THIS ENTRY EXCEEDS CAPACITY OF THE CHAIN,PRINT
C         ERROR MESSAGE AND QUIT
      IF(NENTRY.GT.NFEMAX) GO TO 90
C         PLACE TRANSACTION IN FIRST AVAILABLE LOCATION
   20 K=ILOCFE(NENTRY)
      NLOC=K
      JTIME(K)=KTIME
      JTYPE(K)=KTYPE
      JFSN(K)=KFSN
      JQTY(K)=KQTY
      JPRIOR(K)=KPRIOR
C         CHECK TO SEE IF THIS IS THE ONLY ENTRY ON THE LIST
      IF(NENTRY.NE.1) GO TO 40
C         UPDATE LIST STATUS AND POINTER VARIABLES
      NFIRST=K
      NTIME=KTIME
      JPOINT(K)=0
      GO TO 100
C         DOES THE NEW TRANSACTION PRECEED ALL OTHER LIST ENTRIES?
   40 IF(KTIME.GE.NTIME) GO TO 60
C         INSERT NEW TRANSACTION AS A FIRST LINK IN THE CHAIN
      NTIME=KTIME
      JPOINT(K)=NFIRST
      NFIRST=K
      GO TO 100
C         THE FOLLOWING STEPS SEARCH DOWN THE FEL AND
C         INSERTS THE TRANSACTION IN THE PROPER LOCATION
   60 JJ=NFIRST
      KJ=JPOINT(JJ)
C         IS JJ THE LAST RECORD ON THE CHAIN?
   62 IF(KJ.EQ.0) GO TO 97
C         DOES THE NEW TRANSACTION PRECEED KJ?
```

ENTER
p 2 of 2

```
      IF(KTIME.LT.JTIME(KJ)) GO TO 80
      JJ=KJ
      KJ=JPOINT(JJ)
      GO TO 62
C        INSERT NEW TRANSACTION AS THE LAST LINK ON THE CHAIN
   97 JPOINT(JJ)=K
      JPOINT(K)=0
      GO TO 100
C        INSERT NEW TRANSACTION ON THE CHAIN
   80 JPOINT(JJ)=K
      JPOINT(K)=KJ
      GO TO 100
C        WRITE ERROR MESSAGE
   90 WRITE(6,91)
   91 FORMAT(1H0,55HCAPACITY OF FE CHAIN EXCEEDED CHAIN LISTED ON NEXT P
     CAGE)
      WRITE(6,92)
   92 FORMAT(1H1,33X,66HK          JTIME       JTYPE      JPOINT       JFSN
     C JQTY        JPRIOR,/)
      WRITE(6,93) (I,JTIME(I),JTYPE(I),JPOINT(I),JFSN(I),JQTY(I),I=1,NFE
     CMAX)
   93 FORMAT(1H ,31X,I4,4X,I8,I9,5X,I5,3X,I10,I9,4X,I2)
      RETURN
  100 CONTINUE
      IF(IDBUG.NE.1) GO TO 25
      WRITE(6,8000)ITIN,KTIME,KTYPE,KFSN,KQTY,KPRIOR,K,NFIRST,NENTRY,NT
     CIME
 8000 FORMAT(7H ENTER ,I10,40X,8I6,I7)
   25 CONTINUE
      RETURN
      END
```

Subroutine: ENTERB

Function:

This routine records backorders in the backorder file, and updates associated pointed variables.

Description:

This routine records backorders in the backorder file. Backorders are stored in link list form, with pointers from each recorded backorder to the oldest outstanding backorder for the associated item. In this link list, the highest priority, oldest backorder is stored first; that is, backorders are recorded on a first in first out basis within priority. See Volume I, Chapter V for variable definitions.

The variable NBOPT (N) defines the location of the oldest, highest priority backorder for item N. When new stock is received, this backorder will be the first requisition to be filled.

ENTERB
p. 1 of 2

```
1   04-16-77    12.104
          SUBROUTINE ENTERB(N,IQTY,IPRI,JTIME)
C           THIS ROUTINE BACKORDERS RFQUISITIONS FOR ITEM N
C             IQTY=QUANTITY PLACED ON BACKORDER
C             IPRI =1  HIGH-PRIORITY RFQUISITION
C             IPRI=2  OTHERWISE
C             JTIME= CLOCK TIME RFQ WAS RECEIVED
          COMMON/NBMAX/NBMAX
          COMMON/NLOCBK/NLOCBK
          COMMON/NBOPT/NBOPT(50)
          COMMON/IBACPT/IBACPT(200)
          COMMON/IDFSNB/IDFSNB(200)
          COMMON/ILOCBK/ILOCBK(200)
          COMMON/IPRIOR/IPRIOR(200)
          COMMON/IQTYB/IQTYB(200)
          COMMON/ITMBAC/ITMBAC(200)
C           RESERVE A STORAGE LOCATION FOR THIS INFORMATION
          IPT=ILOCBK(NLOCBK)
          NLOCBK=NLOCBK-1
C           DID THIS REQUISITION CAUSE THE BACKORDER FILE TO OVERFLOW
          IF(NLOCBK.GE.0)GO TO 20
C           WRITE ERROR MESSAGE
          WRITE(6,91)
       91 FORMAT(1H0,20X,53HERROR--BACKORDER FILE OVERFLOW,FILE DUMP ON NEXT
         C PAGE)
          WRITE(6,82)
       82 FORMAT(1H1,10X,23H**BACKORDFR FILE DUMP**)
          DO 83 K=1,NBMAX
       83 WRITE(6,84) K,ITMBAC(K),IDFSNB(K),IPRIOR(K),IQTYB(K),IBACPT(K)
       84 FORMAT(1H ,3X,7HREC NO=,I3,5X,7HITMBAC=,I7,5X,7HIDFSNB=,I10,5X,7HI
         CPRIOR=,I1,5X,6HIQTYB=,I7,5X,7HIBACPT=,I7)
          RETURN
C           RECORD QUANTITY,PRIORITY,FSN ID,AND TIME DATA FOR THIS BO REQ
       20 ITMBAC(IPT)=JTIME
          IDFSNB(IPT)=N
          IPRIOR(IPT)=IPRI
          IQTYB(IPT)=IQTY
C           ARE ANY OTHER BACKORDERS OUTSTANDING ON ITEM N
          IF(NBOPT(N).GT.0) GO TO 40
C           RECORD POINTER DATA
          NBOPT(N)=IPT
          IBACPT(IPT)=0
          RETURN
C           IS THE NEW BO A PRIORITY 1 REQUISITION
       40 IF(IPRI.EQ.1) GO TO 60
C           NOTE-- NEW LOW PRIORITY BACKORDERS ARE INSERTED LAST ON THE
C                  BACKORDER CHAIN. THE REMAINING STEPS IN THIS PORTION
C                  OF THE SUBROUTINE ACCOMPLISH THIS OBJECTIVE
C           SET JPT EQUAL TO THE FILE LOCATION NO OF THE FIRST BACKORDERED
C           REQUISITION IN THE CHAIN
          JPT=NBOPT(N)
C           IS JPT THE LAST LINK IN THE CHAIN
       49 IF(IBACPT(JPT).EQ.0) GO TO 50
```

ENTER B,
p. 2

```
      KPT=IBACPT(JPT)
      JPT=KPT
      GO TO 49
C        RECORD NEW POINTERS
   50 IBACPT(JPT)=IPT
      IBACPT(IPT)=0
      RETURN
C        SET JPT EQUAL TO LOCATION NO OF FIRST BO ON CHAIN
   60 JPT=NBOPT(N)
C        IS BACKORDER JPT A HIGH PRI BO
   61 IF(IPRIOR(JPT).NE.1) GO TO 80
C        IS JPT THE LAST BACKORDER ON THE CHAIN
      IF(IBACPT(JPT).EQ.0) GO TO 62
      KPT=JPT
      JPT=IBACPT(KPT)
      GO TO 61
C        INSERT NEW BO AS LAST LINK ON CHAIN
   62 IBACPT(JPT)=IPT
      IBACPT(IPT)=0
      RETURN
C        IS JPT THE ONLY BO ON CHAIN
   80 IF(JPT.NE.NBOPT(N)) GO TO 81
C        INSERT NEW BO AS FIRST LINK ON CHAIN
      NBOPT(N)=IPT
      IBACPT(IPT)=JPT
      RETURN
C        INSERT NEW BO AS LINK BETWEEN KPT AND JPT
   81 IBACPT(KPT)=IPT
      IBACPT(IPT)=JPT
      RETURN
      END
```

Subroutine: FILLBO

Function:

This routine is called to initiate shipping actions to fill outstanding backorders.

Description:

Subroutine FILLBO (N) is called to initiate shipping actions to fill outstanding backorders for item N. The routine assumes that backordered requisitions are filled using a first in first out, by priority, issue rule. If on hand assets are insufficient to completely fill a given requisition, partial shipments are initiated. Reduced shipments to partially fulfill several outstanding backorders are not permitted in this routine.

When FILLBO is called, it tests if the highest priority outstanding requisition may be filled completely from on hand stock. If so, and if this shipment will not take on hand stock below the support level, the requisition is removed from the backorder list, and a shipment for the requisition quantity is initiated. If the backorder is low priority (that is, code 2), shipments will be made until on hand stocks just equal the support level ISUL (N) for item N. For high priority requisitions, shipments are made until on hand stock is reduced to zero or until all high priority requisitions are filled, whichever occurs first.

FILLBO,
p. 1 of 2

```
04-16-77    12.096
      SUBROUTINE FILLBO(N)
      COMMON/IDBUG/IDBUG
      COMMON/NBOPT/NBOPT(50)
      COMMON/INVACT/INVACT(50)
      COMMON/IQTYB/IQTYB(200)
      COMMON/ISUL/ISUL(50)
      COMMON/IPRIOR/IPRIOR(200)
      COMMON/ISHIPI/ISHIPI(30,3)
      COMMON/NBOTU/NBOTU(50)
      COMMON/ISHIPT/ISHIPT(30,3)
      COMMON/IBACPT/IBACPT(200)
      COMMON/NLOCBK/NLOCBK
      COMMON/ILOCBK/ILOCBK(200)
   25 IF(NBOPT(N).LE.0) RETURN
      IF(IDBUG.NE.1) GO TO 27
       I=NBOPT(N)
       WRITE(6,990)N,INVACT(N),IQTYB(I),IPRIOR(I),I
  990  FORMAT(' ****FILLBO--ITEM=',I5,'  OH=',I5,'  IQTYB=',I5,
     &         '  IPRIOR=',I5,'  IPT=',I5)
   27 CONTINUE
C        SET IPT EQUAL TO THE POINTER NUMBER OF THE OLDEST,HIGHEST
C        PRIORITY REQUISITION FOR ITEM N ON BACKORDER STATUS.
      IPT=NBOPT(N)
       IF(INVACT(N).LE.0)RETURN
C     **** NOTE ****
C      (A) THIS ROUTINE ASSUMES BO REQUISITIONS ARE FILLED USING
C          A FIFO,BY PRIORITY,ISSUE RULE
C      (B) IF ON-HAND ASSETS ARE INSUFFICIENT TO COMPLETELY FILL A
C          GIVEN REQUISITION,PARTIAL SHIPMENTS ARE INITIATED.
C      (C) REDUCED SHIPMENTS TO PARTIALLY FULFILL SEVERAL
C          OUTSTANDING BACKORDERS ARE NOT PERMITTED IN THIS ROUTINE
C
C        SET ITEST EQUAL TO THE REMAINING ON-HAND INVENTORY,
C        IF THE REQUISITION WAS TO BE COMPLETELY FILLED
       ITEST=INVACT(N)-IQTYB(IPT)
C      IS ITEST ABOVE THE SUPPORT LEVEL
      IF(ITEST.GT.ISUL(N)) GO TO 80
C        IS THIS A PRIORITY 1 REQUISITION
      IF(IPRIOR(IPT).NE.1) GO TO 60
C        CAN THE REQ BE COMPLETELY SATISFIED FROM ON HAND STOCK
      IF(INVACT(N).GE.IQTYB(IPT)) GO TO 80
C        SHIP ALL REMAINING ON-HAND STOCK IN PARTIAL FULFILLMENT
C        OF THIS REQUISITION
      IQTYS=INVACT(N)
C        UPDATE PRIORITY 1 BACKORDER STATISTICS
      JQTYS=-IQTYS
      CALL CUM(ISHIPI,IQTYS,N)
      GO TO 65
C        ARE ON-HAND ASSETS ABOVE THE SUPPORT LEVEL
   60 IF(INVACT(N).LE.ISUL(N)) GO TO [illegible]
C        SHIP DOWN TO THE SUPPORT LEVEL
      IQTYS=INVACT(N)-ISUL(N)
```

FILLBO, p.2

```
C         UPDATE STOCK STATUS RECORDS TO REFLECT THE PARTIAL SHIPMENT
     65 INVACT(N)=INVACT(N)-IQTYS
        NBOTU(N)=NBOTU(N)-IQTYS
        IQTYB(IPT)=IQTYB(IPT)-IQTYS
C         UPDATE GROSS PERFORMANCE STATISTICS
        JQTYS=-IQTYS
        CALL CUM(ISHIPT,IQTYS,N)
C         AT THIS POINT,NO FURTHER SHIPMENTS SHOULD BE POSSIBLE, SINCE
C         HIGH PRIORITY REQUISITIONS ARE PROCESSED FIRST
     70 RETURN
C         SHIP TO FILL THE ENTIRE REQUISITION
     80 IQTYS=IQTYB(IPT)
C         UPDATE STOCK STATUS RECORDS
        INVACT(N)=INVACT(N)-IQTYS
        NBOTU(N)=NBOTU(N)-IQTYS
        JQTYS=-IQTYS
        CALL CUM(ISHIPT,IQTYS,N)
C         IS THIS A PRIORITY 1 REQ.
        IF(IPRIOR(IPT).NE.1) GO TO 100
C         UPDATE PRIORITY 1 STATISTICS
        CALL CUM(ISHIPI,IQTYS,N)
C         REMOVE REQUISITION FROM THE BACKORDER FILE
    100 NBOPT(N)=IBACPT(IPT)
        NLOCBK=NLOCBK+1
        ILOCBK(NLOCBK)=IPT
C         RETURN TO BEGINNING OF ROUTINE TO SEE IF ANY MORE BO
C         NEED TO BE PROCESSED.
        GO TO 25
        END
```

Subroutine: FOR576

Function:

This routine updates forecasted demand rates and standard deviation estimates based on information recorded in the History File.

Description:

This routine uses an eight quarter moving average to update forecasts of the net annual demand rate ADR(N), the MAD of quarterly unit demand QMAD, and average requisition size REQSIZ(N) associated with item N. The estimate of the standard deviation of lead time demand RSIGLT(N) is then updated using the formula

```
RSIGLT(N)=0.5945*QMAD*(0.82375+0.42625*RLT)
```

where RLT denotes the total of the administrative and production leadtimes for item N.

To avoid numerical problems in other routines, FOR576 compares the computed values to the following bounds, and, if the bound is violated, FOR576 resets the variable equal to the associated bound.

$$\begin{array}{rcl} \mathrm{ADR(N)} & \geq & 0.0 \\ \mathrm{QMAD} & \geq & .01 \\ \mathrm{RSIGLT(N)} & \geq & .01 \\ \mathrm{REQSIZ(N)} & \geq & 1.0 \end{array}$$

FOR576

1 of 3

```
$      IDENT   WP0287,ADDSC-CALLAHAN
$      FORTY   DECK
$      PRMFL   C*,W,S,REQS/FOR576.0
$      REMOTE  P*,AC
$      REMOTE  $$,AC
      SUBROUTINE FOR576(N)
       COMMON/IFBUG/IFBUG
       COMMON/ITIME/ITIME
       COMMON/ITQTR/ITQTR
       COMMON/NDEMAC/NDEMAC(1)
       COMMON/NRETAC/NRETAC(1)
       COMMON/NREQAC/NREQAC(1)
      COMMON/NDEMND/NDEMND(1,24)
       COMMON/REQSIZ/REQSIZ(1)
      COMMON/NRETUR/NRETUR(1,24)
       COMMON/NREQ/NREQ(1,24)
      COMMON/ADR/ADR(1)
      COMMON/RSIGLT/RSIGLT(1)
      COMMON/NDENT/NDENT(1)
      COMMON/LTADM/LTADM(1)
      COMMON/LTPROD/LTPROD(1)
      COMMON/UCOST/UCOST(1)
      COMMON/IDBUG/IDBUG
      COMMON/ICDFOR/ICDFOR
      COMMON/ICDSIG/ICDSIG
      COMMON/SRFOR/SRFOR,SRPLT,GDFOR
       LW=6
       NLOOP=24
      IDUM=0
       IFUM=0
      IRUM=0
       KK= NDENT(N)
       IF(KK.GT.8)KK=8
       KL=KK-1
       DO 1 I=1,KL
      IDUM=IDUM + NDEMND(N,I)
       IFUM=IFUM + NREQ(N,I)
    1 IRUM=IRUM + NRETUR(N,I)
C
       IT=MOD(ITIME,ITQTR)
       R=FLOAT(IT)/FLOAT(ITQTR)
       RM=1.-R
       DUM=R*FLOAT(NDEMAC(N))+FLOAT(IDUM)+RM*FLOAT(NDEMND(N,KK))
       FUM=R*FLOAT(NREQAC(N))+FLOAT(IFUM)+RM*FLOAT(NREQ(N,KK))
       RUM=R*FLOAT(NRETAC(N))+FLOAT(IRUM)+RM*FLOAT(NRETUR(N,KK))
```

FOR576 (CONT)

2 of 3

```
C
C
*
*                 COMPUTE LEVELS BASED ON NET DEMANDS, WHERE NET IS
*                      DEMAND MINUS SERVICEABLE RETURNS FOR THE PA
*                      MONTHS,
*                 RETURNS ARE ADDITIONS TO ON-HAND ASSETS.
*
          FOR=DUM-RUM
          IF(FOR.LE.0.)GO TO 1000
          FORCST=FOR/FLOAT(KK)
    900   ADR(N)=FORCST*4.
C
C                 COMPUTE MAD OF QUARTERLY DEMAND
C
          QMAD=0.
          FORCST=DUM/FLOAT(KK)
          DO 20 I=1,KK
          ABSDEV=ABS(FLOAT(NDEMND(N,I))-FORCST)
    20    QMAD=QMAD+ABSDEV
          QMAD=QMAD/FLOAT(KK)
          IF(QMAD.LT. 0.01) QMAD=0.01
C
C                 COMPUTE STANDARD DEVIATION OF LEADTIME DEMAND
C
          RLT=FLOAT(LTADM(N)+LTPROD(N))
          RSIGLT(N)=0.5945*QMAD*(0.82375+0.42625*RLT)
          IF(RSIGLT(N).LT. 0.01) RSIGLT(N)= 0.01
C
```

```
C              COMPUTE AVERAGE REQUISITION SIZE
C
        IF(IFUM.LT.1)IFUM=1
        R=FLOAT(IDUM)/FLOAT(IFUM)
        IF(R.LT.1.)R=1.
        REQSIZ(N)=R
  2000 IF(IFBUG .NE. 1) RETURN
       WRITE(LW,100)
       WRITE(LW,200) N,NDENT(N),NLOOP,ICDFOR,ICDSIG,LTADM(N),LTP
       WRITE(LW,300)
        WRITE(LW,200)NDEMAC(N),(NDEMND(N,J),J=1,KK)
        WRITE(LW,200)NRETAC(N),(NRETUR(N,J),J=1,KK)
        WRITE(LW,200)NREQAC(N),(NREQ(N,J),J=1,KK)
       WRITE(LW,400)
       WRITE(LW,500) FORCST,ADR(N),REQSIZ(N),UCOST(N),QMAD,RSIGL
   100  FORMAT(1H ,4X,'****FOR576--N,NDENT,NLOOP,ICDFOR,ICDSIG',
&                 LTADM,LTPROD')
   200 FORMAT(1H ,4X,12I7/12I7)
   300  FORMAT(1H ,9X,'DEMAND, RETURNS, AND FREQ',
&               IN CURRENT HISTORY FILE')
   400  FORMAT(1H ,'    FORCST       ADR   AVE-REQ    UCOST',
&                     QMAD     RSIGLT')
   500  FORMAT(1H ,6F10.4)
       RETURN
  1000 RSIGLT(N)=.5
       ADR(N)=0.0
       IF(IFBUG .NE. 1) RETURN
       GO TO 2000
       END
$       ENDJOB
```

Subroutine: FORUPD

Function:

This routine slides the demand history files for all items at the end of each quarter. It also resets the demand accumulator variables.

Description:

Subroutine FORUPD first increments the counter NDENT(N) for each item N. This variable counts the number of periods that item N has been in the inventory system.

The routine then slides the history values to the next oldest time slot. Variables affected are the NDEMND, number of units demanded per quarter, NRETUR, the number of serviceable returns, and NREQ, the number of requisitions submitted. The oldest quarter of data is dropped. The accumulator variables NDEMAC(N), NREQAC(N), and NRETAC(N) are then reset to zero. These variables are used to record the cumulative demand activity that occurs within the current quarter.

Finally, FORUPD places a new type 9 event on the Future Events List. This causes FORUPD to be called again IDFOR time units in the future.

```
$      IDENT   WP0287,ADDSC-CALLAHAN
$      FORTY   DECK
$      PRMFL   C*,W,S,REQS/FORUPD.O
$      REMOTE  P*,AC
$      REMOTE  $$,AC
      SUBROUTINE FORUPD
C       THIS ROUTINE SLIDES THE DEMAND HISTORY FILES FOR ALL IT
C       THE END OF EACH FORECAST INTERVAL, RESETS IDAY, AND ZERO
C       THE DEMAND ACCUMULATORS
      COMMON/ITFOR/ITFOR,IDFOR
      COMMON/NITEM/NITEM
       COMMON/NDEMND/NDEMND(1,24)
       COMMON/NRETUR/NRETUR(1,24)
       COMMON/NREQ/NREQ(1,24)
       COMMON/NDEMAC/NDEMAC(1)
       COMMON/NRETAC/NRETAC(1)
       COMMON/NREQAC/NREQAC(1)
       COMMON/NDENT/NDENT(1)
      N=1
C       HAS ITEM N ENTERED THE SYSTEM YET
    5  CONTINUE
       NDENT(N)=NDENT(N)+1
       IF(NDENT(N).GE.0) GO TO 20
      GO TO 25
C    *** NOTE ***
C       BY CONVENTION NDENT(N) IS NEGATIVE UNTIL THE ITEM ENTERS
C       THE INVENTORY MANAGEMENT SYSTEM
C
C       SLIDE THE DEMAND HISTORY FILES FOR ALL ITEMS
   20 CONTINUE
       KK=8
       KP=KK+1
       KL=KK-1
       DO 21 I=1,KL
       IS1= KP-I
       IS2= KK-I
       NDEMND(N,IS1)=NDEMND(N,IS2)
       NRETUR(N,IS1)=NRETUR(N,IS2)
       NREQ(N,IS1)=NREQ(N,IS2)
  21   CONTINUE
       NREQ(N,1)=NREQAC(N)
       NREQAC(N)=0
       NRETUR(N,1)=NRETAC(N)
       NRETAC(N)=0
      NDEMND(N,1)=NDEMAC(N)
      NDEMAC(N)=0
C       HAVE ALL ITEMS BEEN PROCESSED
```

```
   25 IF(N.GE.NITEM) GO TO 26
      N=N+1
      GO TO 5
   26 CONTINUE
C       CREATE EVENT TO TRIGGER NEXT UPDATE
      ITFOR=ITFOR+IDFOR
      CALL ENTER(ITFOR,9,0,0,0)
      RETURN
      END
$      ENDJOB
```

Subroutine: GP

Function:

This routine provides a X-Y plot of the vectors provided as calling parameters. It is used by subroutine PLOTR to provide four plots of inventory characteristics as a function of time.

Description:

This routine automatically determines appropriate values for maximum and minimum of each plot axis. Definitions of each of the calling parameters are provided on page 1 of the program listing, and will not be discussed further here.

GP
p. 1 of 3

```
      SUBROUTINE GP(X,Y,L,S,M,N,W,LN,A,PLOT)
C**
C**   CONTROL
C**
C
C**   CALL GP (X. Y. L. S, M. W, LN, A, PLOT)
C**
C**   WHERE
C**          X = ARRAY OF INDEPENDENT VALUES, DIMENSIONED X(M).
C**          Y = ARRAY OF SETS OF DEPENDENT VALUES, DIMENSIONED Y(M,N).
C**          L = NUMBER OF LINES TO BE SKIPPED BEFORE DISPLAY.
C**          S = NUMBER OF SPACES FROM THE LEFT SIDE OF PAGE TO
C**              BE SKIPPED BEFORE DISPLAY.
C**          M = NUMBER OF POINTS IN EACH SET.
C**          N = NUMBER OF SETS OF POINTS.
C**          W = WIDTH OF DISPLAY IN PRINT SPACES.
C**         LN = LENGTH OF DISPLAY IN PRINT LINES.
C**          A = ARRAY OF SINGLE CHARACTERS, DIMENSIONED A(N),TO
C**              REPRESENT THE TREND FOR EACH SET (EX.- DATA A/1HA,
C**              1HB,..,ETC.)
C**       PLOT = ARRAY OF SINGLE CHARACTERS GENERATED BY GP TO
C**              DISPLAY TRENDS, DIMENSIONED PLOT (LN,W).
C**
      INTEGER S, W, W1
C
      DIMENSION X(M), Y(M,N), A(N), PLOT(LN,W)
C
      DATA BLANK/1H /, EDGE/1H*/
      DATA ORD/1H /,ABSIS/1H-/
C
C**   CHECK MAXIMUM WIDTH AND LENGTH REQUESTED AND
C**   EXIT IF NOT CORRECT
C**
      IF (S+W .GT. 131) GO TO 900
      IF (L+LN .GT. 58) GO TO 800
C
C**   FIND MINIMUM AND MAXIMUM OF X AND Y
C**
      XMAX=X(1)
      XMIN=X(1)
C
      DO 10 I=2,M
      IF (X(I) .GT. XMAX) XMAX=X(I)
   10 IF (X(I) .LT. XMIN) XMIN=X(I)
C
      YMAX=Y(1.1)
      YMIN=Y(1.1)
C
      DO 20 I=1,M
      DO 20 J=1,N
      IF (Y(I,J) .GT. YMAX) YMAX=Y(I.J)
```

GP
p.2 of 3

```
   20 IF (Y(I,J) .LT. YMIN) YMIN=Y(I,J)
C
C
C**************************************************************
C
C          SET ORIGIN AT (0.,0.)
C
      YMIN=0.
      XMIN=0.
C
C          BEGIN ON NEW PAGE
C
      WRITE(5,610)
  610 FORMAT(1H1)
C**************************************************************
C
C**   COMPUTE SCALE FACTOR -- P FOR X, Q FOR Y
C**
      P=FLOAT(W-1)/(XMAX-XMIN)
      Q=FLOAT(LN-1)/(YMAX-YMIN)
C
C**   BLANK PLOT ARRAY
C**
      DO 30 I=1,W
      DO 30 J=1,LN
   30 PLOT(J,I)=BLANK
C
C**   CONSTRUCT BORDER OF DISPLAY
C**
      DO 40 J=1,LN
      I=1
      PLOT(J,I)=EDGE
      I=W
   40 PLOT(J,I)=EDGE
C
      W1=W-1
      DO 50 I=2,W1
      J=1
      PLOT(J,I)=EDGE
      J=LN
   50 PLOT(J,I)=EDGE
C
C**   COMPUTE SUBSCRIPTS AND INSERT TREND CHARACTER IN
C**   PLOT ARRAY
C**
      DO 60 I=1,M
      DO 60 J=1,N
      I1=1+INT(0.5+P*(X(I)-XMIN))
      J1=LN-INT(0.5+Q*(Y(I,J)-YMIN))
   60 PLOT(J1,I1)=A(J)
C
C**   SKIP L LINES BEFORE BEGINNING DISPLAY PRINTING
```

GP
p.3 of 3

```
C
      DO 70 K=1,L
   70 WRITE (6.600)
  600 FORMAT (1H )
C
C**   WRITE OUT PLOT ARRAY, SKIPPING S SPACES BEFORE PRINTING
C**   EACH LINE OF DISPLAY
C**
      DO 80 J=1,LN
   80 WRITE(5,601) (BLANK,K=1,S),(PLOT(J,I),I=1,W)
  601 FORMAT (132A1)
      WRITE (6.602) XMIN,XMAX,YMIN,YMAX
  602 FORMAT (1H0,5X,6HXMIN =E16.8,10X,6HXMAX =E16.8,10X,
     X      6HYMIN =E16.8,10X,6HYMAX =E16.8)
      RETURN
C**
C**   ERROR MESSAGES BEFORE TERMINATION
C**
  800 WRITE (6.603) L, LN
  603 FORMAT (30HAL+LN IS GREATER THAN 58   L =I3,5X,4HLN =I3)
  900 WRITE (6.604) S, W
  604 FORMAT (30HAS+W IS GREATER THAN 131   S =I3,5X,3HW =I3)
      STOP
      END
```

Subroutine: GUIDE

Function:

This routine establishes guidelines for levels computations to be performed by subroutine LEVELN.

Description:

This routine establishes guidelines for levels computations based on a comparison of current obligations and budget authorizations for the current budget period. However, no buys are permitted after the obligation ceiling for a given budget interval has been reached.

Methodology to be used in establishing buy guidelines is determined by the code ICDGID provided as input to the simulation.

At the end of this event, GUIDE places a new type 7 event on the Future Events List.

GUIDE
p.1 of 2

```
      SUBROUTINE GUIDE
C        THIS ROUTINE ESTABLISHES GUIDELINES FOR LEVELS COMPUTATIONS
C        ACCORDING TO POLICY STATED IN AFLCM 57-6, HOWEVER NO BUYS ARE
C        PERMITTED AFTER THE OBLIGATION CEILING FOR A GIVEN BUDGET
C        INTERVAL IS REACHED.
      COMMON/IGBUG/IGBUG
      COMMON/IDDIV/IDDIV
      COMMON/IOBLIG/IOBLIG
      COMMON/ITDIV/ITDIV
      COMMON/ITHQ/ITHQ
      COMMON/ITIME/ITIME
      COMMON/ITYEAR/ITYEAR
      COMMON/MAXBUY/MAXBUY
      COMMON/NBPER/NBPER
      COMMON/GSLF/GSLF
      COMMON/GRLF/GRLF
      COMMON/GTLF/GTLF
      COMMON/GUIDI/ICDGID,NPH,NPOLCY,IDJAXR,IDJAXS,SHORT(10)
      COMMON/IBAUTH/IBAUTH(12)
      COMMON/GEOQ/GEOQ(3)
C        DO CURRENT OBLIGATIONS EXCEED THE CEILING
      IF(IOBLIG.LE.IBAUTH(NBPER))GO TO 11
      IF(MAXBUY.LE.0) GO TO 20
      MAXBUY=0
      GO TO 20
   11 CONTINUE
      MAXBUY=9999999
   20 CONTINUE
      IF(IGBUG.NE.1) GO TO 30
      R=FLOAT(ITIME)/FLOAT(ITYEAR)
      WRITE(6,8000)ITIME,NBPER,IOBLIG,IBAUTH(NBPER),R,MAXBUY
 8000 FORMAT(7H GID576,I10,5H-----,6HNBPER=,I5,5X,12HOBLIGATIONS=,I10,5X
     C,11HAUTHORIZED=,I10,5X,5HYEAR=,F6.3,5X,8HMAXBUY= ,I9)
   30 CONTINUE
      IF(ICDGID.EQ.1) GO TO 50
      IF(ICDGID.EQ.2) GO TO 40
      GEOQ(1)=3.
      GEOQ(2)=6.
      GEOQ(3)=12.
      GSLF=1.
      GSULF=0.
      GTLF=24.
      GRLF=12.
      GO TO 50
   40 CONTINUE
      CALL A22JAX
   50 CONTINUE
C        COMPUTE TIME FOR NEXT GUIDELINE REVIEW
      ITDIV=ITDIV+IDDIV
C        IF NEXT REVIEW TIME PRECEEDS NEXT BUDGET REVISION
C        PLACE REVIEW EVENT ON FE LIST
      IF(ITDIV.GE.ITHQ) GO TO 13
      CALL ENTER(ITDIV,7,0.0,0)
   13 CONTINUE
      RETURN
      END
```

Subroutine: INBUDG1

Function:

This routine performs initialization functions required in simulating buy guidelines events.

Description:

The subroutine INBUDG1 sets authorized and projected obligation ceilings. The routine also initializes obligation and period counters, and places the initial budget review event on the Future Events List.

The routine sets the authorized budget ceilings equal to the total dollar demand to be generated in the corresponding budget interval. That is, it sets the budget authorization equal to the dollar value of unit demand in the array IDEMND; this array drives the simulated demand activity for the simulation.

Finally, routine INBUDG1 places a type 8 event on the Future Events List.

INBUDG1
p.1 of 2

```
      SUBROUTINE INBDG1
C       THIS ROUTINE SETS AUTHORIZED AND PROJECTED OBLIGATIONS CEILINGS
C       EQUAL TO THE TOTAL DOLLAR DEMAND OBSERVED IN THE CORRESPONDING
C       BUDGET INTERVAL. THE ROUTINE ALSO INITIALIZES OBLIGATION AND
C       PERIOD COUNTERS AND PLACES THE BUDGET REVIEW EVENT ON THE FEL.
      DIMENSION TOTAL(30)
      COMMON/BPER/BPER
      COMMON/IOBLIG/IOBLIG
      COMMON/NITEM/NITEM
      COMMON/INQTR/INQTR
      COMMON/NBPER/NBPER
      COMMON/ITHQ/ITHQ
      COMMON/IDEMND/IDEMND(50,81)
      COMMON/UCOST/UCOST(50)
      COMMON/IBAUTH/IBAUTH(12)
      COMMON/IBPROJ/IBPROJ(12)
C
C          COMPUTE ACTUAL $/QTR DEMAND
C
      ISW=1
      IF(ISW.EQ.1)GO TO 200
      DO 30 I=1,27
      TOTAL(I)=0.
      SUM=0.
      DO 13 N=1,NITEM
      IF(IDEMND(N,I).GE.999999) GO TO 13
      SUM=SUM+FLOAT(IDEMND(N,I))*UCOST(N)
   13 CONTINUE
      TOTAL(I)=SUM
   30 CONTINUE
C
C          SET AUTHORIZED BUDGET EQUAL TO ACTUAL DEMAND
C
      DO 50 I=1,7
      K=4*I-3
      SUBTOT=TOTAL(K)+TOTAL(K+1)+TOTAL(K+2)+TOTAL(K+3)
      IBAUTH(I)=BPER*SUBTOT
      IBPROJ(I)=SUBTOT
   50 CONTINUE
      WRITE(6,8000)(I,TOTAL(I),I=1,27)
 8000 FORMAT(///20X,6HQTR NO.8X,7H$DEMAND///30(I26,F15.2/))
      WRITE(6,8010)(K,IBPROJ(K),IBAUTH(K),K=1,7)
 8010 FORMAT(///20X,6H YR NO.8X,7H$ACTUAL.10X,5H$AUTH///10(I26.2I15/))
      GO TO 500
  200 CONTINUE
      DO 250 I=1,7
      IBAUTH(I)=99999999.
  250 IBPROJ(I)=99999999.
  500 CONTINUE
C        ENTER BUDGET REVISION EVENT ON FE LIST
      CALL ENTER(ITHQ,8,1,0,0)
C        INITIALIZE BUDGET INTERVAL COUNIER
```

INBUDG1
p. 2 of 2

```
      NBPER=0
      RETURN
      END
```

Subroutine: INFEL

Function:

This routine initializes the Future Events List to an empty status.

Description:

Subroutine INFEL initializes the Future Events List. Subsequently, Subroutines ENTER and REMOVE update this list. See Volume I, Chapter IV, for a detailed discussion of these activities.

INFEL,
p. 1 of 1

```
1   02-11-77     10.687
          SUBROUTINE INFEL
C           THIS ROUTINE INITIALIZES THE FUTURE EVENTS
C           LIST TO AN EMPTY STATUS
          COMMON/NENTRY/NENTRY
          COMMON/NFEMAX/NFEMAX
          COMMON/NTIME/NTIME
          COMMON/ILOCFE/ILOCFE(500)
          NFEMAX=500
          NENTRY=0
          NTIME=9999999
          DO 1 I=1,NFEMAX
        1 ILOCFE(I)=I
          RETURN
          END
```

Subroutine: INITAL

Function:

This routine sets the values of critical timing variables and initializes the Future Events List.

Description:

First, this routine calls Subroutine INFEL to initialize the Future Events List. Next, major parameters for the Backorder File are set, and the Backorder File is initialized.

Major timing variables are then set assuming that there are 100 time units in each simulated day. At present, Subroutine INITAL assumes that there are 7 days per week, 4 weeks per month, 3 months per quarter, and 4 quarters in each year.

Subroutine INITAL then sets the simulation clock and statistics collection variables; specifically, it sets:

ITIME = 0.

ITINV = 1.

INITAL then sets other significant timing variables discussed in Chapter IV, Volume I.

Finally, INITAL places transactions for event types 3, 5, 6, 10, 11, 12, and 13 on the Future Events List, provided these events are to be simulated in the current simulation run.

p 1 of 3

```
$      IDENT   WP0287,ADDSC-CALLAHAN
$      FORTY   DECK
$      PRMFL   C*,W,S,REQS/INITAL.0
$      REMOTE  P*,AC
$      REMOTE  $$,AC
      SUBROUTINE INITAL
C       THIS ROUTINE
C       INITIALIZES THE FUTURE EVENTS LIST.
C
       COMMON/IDBUG/IDBUG,IEBUG,IFBUG,IGBUG,IHBUG
       COMMON/ITRACE/ITRACE,ISTRAC
      COMMON/IDDIV/IDDIV
      COMMON/IDLEVL/IDLEVL
      COMMON/IDSTAT/IDSTAT
      COMMON/IDTHQ/IDTHQ
      COMMON/IKDEM/IKDEM
      COMMON/INQTR/INQTR
      COMMON/IQTRND/IQTRND
      COMMON/ISTOCK/ISTOCK,IDSTOC
      COMMON/ISTOP/ISTOP
      COMMON/ISTAT/ISTAT
      COMMON/ITCANB/ITCANB,IDCANB
      COMMON/ITDAY/ITDAY
      COMMON/ITDIV/ITDIV
      COMMON/ITFOR/ITFOR,IDFOR
      COMMON/ITHQ/ITHQ
      COMMON/ITIME/ITIME
      COMMON/ITINV/ITINV
      COMMON/ITLEVL/ITLEVL
      COMMON/ITMNTH/ITMNTH
      COMMON/ITQTR/ITQTR
      COMMON/ITWEEK/ITWEEK
      COMMON/ITYEAR/ITYEAR
      COMMON/MBODAY/MBODAY
      COMMON/NBMAX/NBMAX
      COMMON/NENTRY/NENTRY
      COMMON/NFEMAX/NFEMAX
      COMMON/NFIRST/NFIRST
      COMMON/NITEM/NITEM
      COMMON/NLOC/NLOC
      COMMON/NLOCBK/NLOCBK
      COMMON/NTIME/NTIME
      COMMON/RAND/IX,IY
      COMMON/ILOCBK/ILOCBK(200)
C
```

```
47Ø     C       INITIALIZE THE FUTURE EVENTS LIST
48Ø     C
49Ø            NFEMAX=5ØØ
5ØØ           CALL INFEL
51Ø     C
52Ø     C        SET PARAMETERS FOR BACKORDER FILE
53Ø     C
54Ø            NBMAX=1ØØ
55Ø           NLOCBK=NBMAX
56Ø           DO 1Ø I=1,NBMAX
57Ø        1Ø ILOCBK(I)=NBMAX+1-I
58Ø     C       SET TIMING VARIABLES BASED ON 1ØØ TIME UNITS PER DAY
59Ø     C          7 DAYS/WEEK, 4  WEEKS/MONTH , 4 QUARTERS/YEAR
6ØØ           ITDAY=1ØØ
61Ø           ITWEEK=7*ITDAY
62Ø           ITMNTH=4*ITWEEK
63Ø           ITQTR=3*ITMNTH
64Ø           ITYEAR=4*ITQTR
65Ø     C       SET TIMING VARIABLES FOR MANAGEMENT AND DATA COLLECTION
66Ø     C          CURRENT SIMULATION CLOCK TIME
67Ø           ITIME=Ø
68Ø     C          CURRENT STATISTICS COLLECTION INTERVAL
69Ø           ITINV=1
7ØØ     C          END OF CURRENT QUARTER
71Ø           IQTRND=ITQTR
72Ø     C          TIME OF FIRST HQ USAF BUDGET AUTHORIZATION
73Ø           ITHQ=1Ø
74Ø     C          TIME BETWEEN HQ USAF BUDGET REVISIONS
75Ø           IDTHQ=4*ITQTR
76Ø     C          TIME OF FIRST DIVISION LEVEL REVIEW
77Ø           ITDIV=2Ø
78Ø     C          TIME BETWEEN DIVISION LEVEL REVIEWS
79Ø           IDDIV=ITMNTH
8ØØ     C          TIME OF FIRST STOCK LEVEL COMPUTATION
81Ø           ITLEVL=3Ø
82Ø     C          TIME BETWEEN STOCK LEVEL COMPUTATIONS
Ø83Ø           IDLEVL=2*ITWEEK
84Ø     C        TIME OF FIRST STOCK STATUS REVIEW
85Ø           ISTOCK=4Ø
86Ø     C        TIME BETWEEN STOCK STATUS REVIEQS
Ø87Ø           IDSTOC=2*ITWEEK
88Ø     C         TIME TO ACTIVATE STATISTICS COLLECTION ROUTINE
89Ø           ISTAT=ITWEEK-5
9ØØ     C          TIME BETWEEN STATISTICAL UPDATES
91Ø           IDSTAT=ITWEEK
92Ø     C        STOP AFTER SIMULATION INQTR QUARTERS
93Ø           ISTOP=INQTR*ITQTR
94Ø     C       PLACE INITIAL MANAGEMENT AND DATA COLLECTION EVENTS ON
```

p 3 of 3

```
C       THE FUTURE EVENTS LIST.
C              STAT--REVIEW STOCK STATUS
      CALL ENTER(ISTOCK,5,Ø,Ø,Ø)
C              LEVEL--COMPUTE INV CONTROL LEVELS
      CALL ENTER(ITLEVL,6,Ø,Ø,Ø)
C              SSTAT--ACCUMULATE STATUS STATISTICS
      CALL ENTER(ISTAT,11,1,Ø,Ø)
C              ITØUT--END OF SIMULATION PERIOD
      CALL ENTER(ISTOP,1Ø,Ø,Ø,Ø)
C
C        CREATE FORECAST EVENT
C
  1ØØ CONTINUE
       ITFOR=ITQTR
       IDFOR=ITQTR
C              FORUPD--UPDATE DEMAND HISTORY FILES
      CALL ENTER(ITFOR,9,Ø,Ø,Ø)
C              DEMPAR--GENERATE DEMAND
      CALL ENTER(1ØØ,12,Ø,Ø,Ø)
C
C              CANCLB--CANCELLATION REVIEW EVENT GOES HERE--
C                  CANCELLATIONS ARE NOT SIMULATED IN THIS STUDY
C
       MBODAY=1ØØ
       IDCANB=ITMNTH
C
C        INITIALIZE FOR PLOT OF STOCK HISTORY
C
       IF(ISTRAC.LE.Ø) GO TO 2ØØ
       CALL ENTER(ITRACE,13,Ø,Ø,Ø)
 2ØØ   CONTINUE
      RETURN
      END
$      ENDJOB
```

Subroutine: INITEM

Function:

This routine initializes the Item Data File by reading input from File 07, and then performs item-related initializing calculations.

Description:

Input may be provided to this routine in either binary or BCD format. The specific type of input is determined by the variable INTYPE. A detailed description of data contained on file 07 is presented in Vol I, Chap VII.

After reading information on a given item, INITEM then performs several calculations. First, it zeros the demand history accumulators NRETAC, NDEMAC, and NREQAC used to record demand activity for each given item. Inventory status variables and associated pointers are then updated. In the present coding, all due-in inventories are assumed to be on hand; consequently, INITEM increases the variable INVACT(N) by the amount of due-in inventory, and sets the variable INVDUE(N) to zero.

Next, the demand history arrays are initialized using the first NDHIS periods of data recorded in the Demand Driver File. See Vol I, Chap IV, for definition of these data elements.

Finally, subroutine INITEM initializes the variables IBOPOH(J) and IBOPOR(J). These variables record the initial and on order inventories at the beginning of the simulation run.

If an end of file is encountered in reading File 07, subroutines OUT, OUTCST, and PLOTR are called to write out performance statistics summaries.

```
2Ø     $       IDENT   WPØ287,ADDSC-CALLAHAN
3Ø     $       FORTY   DECK
4Ø     $       PRMFL   C*,W,S,REQS/INITEM.O
5Ø     $       REMOTE  P*,AC
6Ø     $       REMOTE  $$,AC
7Ø            SUBROUTINE INITEM
8Ø     C************INITEM******************
85             CHARACTER ALC,FSN,UM,NOUN,MGTCD
9Ø             COMMON/FSN/ALC,FSN(4),UM,NOUN(2),MGTCD(4),IOH,IOR,IPPL,IR
1ØØ           COMMON/GSLF/GSLF
11Ø           COMMON/IDBUG/IDBUG
12Ø           COMMON/ITDAY /ITDAY
13Ø           COMMON/ITMNTH/ITMNTH
14Ø           COMMON/NITEM/NITEM
15Ø            COMMON/NDEM/NDEM
16Ø            COMMON/NDHIS/NDHIS
17Ø            COMMON/INLU/INLU
18Ø            COMMON/INTYPE/INTYPE
19Ø            COMMON/IEBUG/IEBUG
2ØØ           COMMON/IBOP/IBOPOH(3),IBOPOR(3)
21Ø           COMMON/IDEMND/IDEMND(1,24)
22Ø           COMMON/RMREQS/RMREQS(1)
23Ø           COMMON/INVACT/INVACT(1)
24Ø           COMMON/NORDPT/NORDPT(1)
25Ø           COMMON/NDEMAC/NDEMAC(1)
26Ø            COMMON/NRETAC/NRETAC(1)
27Ø            COMMON/NREQAC/NREQAC(1)
28Ø            COMMON/NDEMND/NDEMND(1,24)
29Ø            COMMON/NRETUR/NRETUR(1,24)
3ØØ            COMMON/NREQ/NREQ(1,24)
31Ø           COMMON/NDENT/NDENT(1)
32Ø           COMMON/INVDUE/INVDUE(1)
33Ø           COMMON/NBOPT/NBOPT(1)
34Ø           COMMON/NBOTU/NBOTU(1)
35Ø           COMMON/REQSIZ/REQSIZ(1)
36Ø           COMMON/REQMAD/REQMAD(1)
37Ø           COMMON/LTPROD/LTPROD(1)
38Ø           COMMON/LTADM/LTADM(1)
39Ø           COMMON/UCOST/UCOST(1)
4ØØ           COMMON/ADR/ADR(1)
41Ø           COMMON/ISUL/ISUL(1)
42Ø            COMMON/IREQ  /IREQ(1,24)
43Ø            COMMON/IRETUR/IRETUR(1,24)
44Ø           COMMON/IRL/IRL(1)
45Ø           COMMON/ITL/ITL(1)
46Ø           COMMON/IROL/IROL(1)
47Ø           COMMON/IRQTY/IRQTY(1)
48Ø           COMMON/RMTBR /RMTBR(1)
49Ø           COMMON/RMEAN/RMEAN(1)
5ØØ           COMMON/RTREND/RTREND(1)
51Ø           COMMON/RMAD  /RMAD(1)
52Ø           COMMON/RERSUM/RERSUM(1)
53Ø           COMMON/KNT   /KNT(1)
```

p. 2 of 4

```
          COMMON/GEOQ/GEOQ(3)
          COMMON/GEOQF/GEOQF(3)
           COMMON/ICDFOR/ICDFOR
C
C
C                 SET NUMBER OF PERIODS OF DATA INPUT
C
           IDPER=NDEM
C
C           READ DEMAND DATA FOR ITEM N FROM LOGICAL UNIT LR
           LR=INLU
           IKNT=0
          DO 100 N=1,NITEM
       10 CONTINUE
C
C                 READ ITEM DATA INPUT FROM FILE LR
C
           IF(INTYPE.EQ.2) GO TO 20
C
C                 READ BCD INPUT
C
           IKNT=IKNT+1
           READ(LR,8000,END=200) ALC,FSN,UM,UCOST(N),NOUN,MGTCD,IOH
     &            LTADM(N),LTPROD(N),IPPL,IPPPR
C
           RIPPPR=FLOAT(IPPPR)/100.
           IF(IEBUG.EQ.1)WRITE(6,8010)IKNT,ALC,FSN,UM,UCOST(N),NOUN
     &            IOH,IOR,LTADM(N),LTPROD(N),IPPL,RIPPPR
           IF(IEBUG.EQ.1)WRITE(6,8015)(I,I=1,10)
C
C                 READ DEMAND,RETURNS, AND REQ-FREQUENCY
C
           READ(LR,8000)(IDEMND(N,J),J=1,IDPER)
           IF(IEBUG.EQ.1)WRITE(6,8020)(IDEMND(N,J),J=1,IDPER)
           READ(LR,8000)(IRETUR(N,J),J=1,IDPER)
           IF(IEBUG.EQ.1)WRITE(6,8030)(IRETUR(N,J),J=1,IDPER)
           READ(LR,8000)(IREQ(N,J),J=1,IDPER)
           IF(IEBUG.EQ.1)WRITE(6,8040)(IREQ(N,J),J=1,IDPER)
           GO TO 50
C
C                 READ BINARY DATA
C
       20  CONTINUE
           IKNT=IKNT+1
           READ(LR)ALC,FSN,IR,UM,UCOST(N),NOUN,MGTCD,IOH,IOR,
     &            LTADM(N),LTPROD(N),IPPL,RIPPPR
           IF(IEBUG.EQ.1)WRITE(6,8010)IKNT,ALC,FSN,UM,UCOST(N),NOUN,
     &            MGTCD,IOH,IOR,LTADM(N),LTPROD(N),IPPL,RIPPPR
           READ(LR)ALC,FSN,IR,IDEMND
           IF(IEBUG.EQ.1)WRITE(6,8020)(IDEMND(N,J),J=1,IDPER)
           READ(LR)ALC,FSN,IR,IRETUR
           IF(IEBUG.EQ.1)WRITE(6,8030)(IRETUR(N,J),J=1,IDPER)
           READ(LR)ALC,FSN,IR,IREQ
           IF(IEBUG.EQ.1)WRITE(6,8040)(IREQ(N,J),J=1,IDPER)
C                 WRITE INPUT DATA TO FILE 09
```

p. 3 of 4

```
C
C
C
 8000 FORMAT(V)
 8010 FORMAT(/I5,1X,A2,1X,A2,A4,A6,A3,1X,A2,F11.2,1X,A6,A4,
&               T53,2A1,A4,A2,2X,2I7,3I6,F5.2)
 8015 FORMAT(T21,10I10)
 8020 FORMAT('IDEMND  DEMAND/QTR',(T21,10I10))
 8030 FORMAT('IRETUR  RETURN/QTR',(T21,10I10))
 8040 FORMAT('IREQ    REQ   /QTR',(T21,10I10))
C
   50 CONTINUE
C                INDICATE ITEM HAS NDHIS PERIODS OF DEMAND HISTORY
C
     NDENT(N)=NDHIS
C
C    ZERO DEMAND HISTORY RECORDS
      NRETAC(N)=0
     NDEMAC(N)=0
      NREQAC(N)=0
     DO 65 J=1,NDHIS
   65 NDEMND(N,J)=IDEMND(N,J)
C       SET INVENTORY DUE-IN TO ZERO
   70 INVDUE(N)=0
     NBOTU(N)=0
     NBOPT(N)=0
     NORDPT(N)=0
C
C                LOAD DEMAND HISTORY ARRAYS
C
      KK=NDHIS
      DO 1 I=1,NDHIS
      NDEMND(N,KK)=IDEMND(N,I)
      NRETUR(N,KK)=IRETUR(N,I)
      NREQ(N,KK)=IREQ(N,I)
      KK=KK-1
    1 CONTINUE
C
C                SLIDE DOWN DATA IN DEMAND ARRAYS
C
      KK=NDHIS+1
      DO 2 I=1,IDPER
      IDEMND(N,I)=IDEMND(N,KK)
      IRETUR(N,I)=IRETUR(N,KK)
      IREQ(N,I)=IREQ(N,KK)
      KK=KK+1
    2 CONTINUE
```

```
C
C
C                ESTABLISH BEGINNING INVENTORY LEVELS
C
        INVACT(N)=IOH+IOR
  1ØØ   CONTINUE
C
C           INITIALIZE GROSS ON-HAND AND ON-ORDER STATISTICS
C
  11Ø CONTINUE
      DO 13Ø N=1,NITEM
      IF(NDENT(N).LT.Ø) GO TO 13Ø
      IF(INVACT(N).LE.Ø) GO TO 12Ø
      IBOPOH(1)=IBOPOH(1)+1
      IBOPOH(2)=IBOPOH(2)+INVACT(N)
      IBOPOH(3)=IBOPOH(3)+IFIX(UCOST(N)*FLOAT(INVACT(N)))
  12Ø CONTINUE
      IF(INVDUE(N).LE.Ø) GO TO 13Ø
       IBOPOR(1)=IBOPOR(1)+1
      IBOPOR(2)=IBOPOR(2)+INVDUE(N)
      IBOPOR(3)=IBOPOR(3)+IFIX(UCOST(N)*FLOAT(INVDUE(N)))
  13Ø CONTINUE
  14Ø CONTINUE
      RETURN
  2ØØ WRITE(6,82ØØ)
 82ØØ FORMAT(1H1,////2ØX,'END OF FILE READINF..................
      CALL OUT
       CALL OUTCST
      CALL PLOTR
      STOP
      END
$      ENDJOB
```

Subroutine: ITRSLT

Function:

This routine writes details of the performance statistics associated with a given item to File 08.

Description:

This routine first converts recorded values of performance statistics to floating point variables stored in the array RID. It then writes a single binary record for this item to File 08. Record lay outs for this record are defined in Vol I, App B. See Vol I, Chap VIII, for an additional discussion of this routine.

ITRSLT

p. 1 of 2

```
C             ITRSLT.O  OUTPUT ITEM PERFORMANCE STATISTICS TO FILE 08
C
      SUBROUTINE ITRSLT
      COMMON/FSN/ALC,FSN(4),UM,NOUN(2),MSTCD(4),IOH,IOR,IPPL,IPPPR
      COMMON/IDENT/IDENT,MRUN,IREPL
      COMMON/INQTR/INQTR
      COMMON/COSORD/COSORD(3)
      COMMON/ISMORD/ISMORD(30,3)
      COMMON/ILGORD/ILGORD(30,3)
      COMMON/POLICY/ICDEOQ,ICDSL
      COMMON/LTADM/LTADM(1)
      COMMON/LTPROD/LTPROD(1)
      COMMON/UCOST/UCOST(1)
      COMMON/IREQT/IREQT(30,3)
      COMMON/IRETRN/IRETRN(30,3)
      COMMON/IBAKDT/IBAKDT(30,3)
      COMMON/IORDER/IORDER(30,3)
      COMMON/INVDAY/INVDAY(30,3)
      COMMON/IFILLT/IFILLT(30,3)
      DIMENSION RJD(16,10)
      DIMENSION RID(16,10)
C
C             ON FIRST REPLICATION, ZERO THE RJD-ARRAY
C
      IF(IREPL.GT.1) GO TO 20
      DO 10 I=1,INQTR
      DO 10 J=1,10
      RJD(I,J)=0.
   10 CONTINUE
   20 CONTINUE
C
C
C             CONVERT INTEGERS TO FLOATING POINT FOR OUTPUT
C
      TDENT=IDENT
      TCDSL=ICDSL
      TRUN=MRUN
      TLT=LTADM(1)+LTPROD(1)
      ROH=IOH+IOR
C
C             COMPUTE DIFFERENCES OF I- AND JD- ARRAYS.
C                 THESE ARE THE OPERATING RESULTS FOR THE CURRENT
C                 ITEM.
C
      DO 100 I=1,INQTR
      RID(I,1)=IREQT(I,1)-RJD(I,1)
      RID(I,2)=IREQT(I,2)-RJD(I,2)
      RID(I,3)=IRETRN(I,2)-RJD(I,3)
      RID(I,4)=IBAKDT(I,1)-RJD(I,4)
      RID(I,5)=IBAKDT(I,2)-RJD(I,5)
      RID(I,6)=IORDER(I,2)-RJD(I,6)
      RID(I,7)=INVDAY(I,2)-RJD(I,7)
```

ITRSLT,
p 2 of 2

```
         RID(I,8)=IFILLT(I,1)-RJD(I,8)
         RID(I,9)=IFILLT(I,2)-RJD(I,9)
         RID(I,10)=COSORD(1)*FLOAT(ISHORD(I,1))+
     &           COSORD(2)*FLOAT(ILGORD(I,1)) - RJD(I,10)
  100    CONTINUE
C
C                WRITE RESULTS TO FILE 08
C
         WRITE(8)IDENT,TCDSL,TRUN,ALC,FSN,MGIGD,UCOST(1),TLT,
     &           ROH,RID
C
C
C                RECORD CURRENT VALUES FOR USE ON NEXT ITEM
C
         DO 300 I=1,INQTR
         DO 300 J=1,13
         RJD(I,J)=RID(I,J)+RJD(I,J)
  300    CONTINUE
         RETURN
         END
```

Subroutine: LEVEL

Function:

This routine computes new inventory control levels for each item being simulated.

Description:

Subroutine LEVEL first calls subroutine FOR576(N) to update estimates of annual demand rate, the standard deviation of demand in the leadtime, and average requisition size. Next, subroutine LEVELN(N) is called to compute new control levels for item N. After new control levels have been computed for each item, LEVEL creates the next level calculation event (event type 6), and puts it in the Future Events List.

```
 1  04-16-77    12.076
            SUBROUTINE LEVEL
            COMMON/COSHRT/COSHRT
            COMMON/IDLEVL/IDLEVL
            COMMON/ITLEVL/ITLEVL
            COMMON/NITEM/NITEM
             COMMON/NDENT/NDENT(1)
            DO 10  N=1,NITEM
      C
      C                  UPDATE ESTIMATES OF ANNUAL DEMAND RATE(ADR),RSIGLT,
      C                      AND AVERAGE REQUISITION SIZE
      C
            CALL FOR576(N)
  1457 DO LOOP INDEX N MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
      C
      C                  COMPUTE NEW INVENTORY CONTROL LEVELS
      C
             CALL LEVELN(N)
  1457 DO LOOP INDEX N MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
        100 CONTINUE
      C
      C                  CREATE NEXT LEVELING EVENT
      C
            ITLEVL=ITLEVL + IDLEVL
            CALL ENTER(ITLEVL,6,0,0,0)
            RETURN
            END
```

Subroutine: LEVELN

Function:

This routine computes new reorder, termination, retention, and support levels for a given item N.

Description:

This routine is controlled by the management codes ICDEOQ, ICDSL, and ICDSLL provided as input through File 05. These codes specify formulas to be used in computing order quantities, establishing safety levels, and limiting safety levels, respectively. The meanings of these codes are defined in detail in Vol I, Chap VII.

After order quantity and safety level values are computed, they are bounded by input parameters EOQMIN, EOQMAX, SLMIN, and SLMAX. The safety level bounds are dependent upon bounding flag code ICDSLL.

With the current coding, safety level is set to be no less than SLMIN months of supply. Also, the safety level is limited to be no more than either (a) three times the standard deviation of demand in the leadtime, or (b) the expected number of units to be demanded in the leadtime, whichever is smaller.

The routine concludes by computing termination, retention, and support levels using the parameters GTLF, GRLF, and GSULF.

LEVELN
p. 1 of 4

```
)1  04-16-77    12.082
          SUBROUTINE LEVELN(N)
   C
   C          THIS ROUTINE COMPUTES REORDER,STOCK OBJECTIVE,RETENTION,
   C                TERMINATION, AND SUPPORT LEVELS.
   C
          COMMON/IHBUG/IHBUG
          COMMON/COSHRT/COSHRT
         COMMON/COSHLD/COSHLD
         COMMON/COSORD/COSORD(3)
          COMMON/CSTBRK/CSTBRK
         COMMON/GSULF/GSULF
         COMMON/GRLF/GRLF
         COMMON/GSLF/GSLF
         COMMON/GTLF/GTLF
         COMMON/ITLEVL/ITLEVL
         COMMON/IDLEVL/IDLEVL
         COMMON/NITEM/NITEM
         COMMON/POLICY/ICDEOQ,ICDSL,EOQMAX,EOQMIN,SLMAX,SLMIN,RLF,TLF,SULF
         COMMON/ADR/ADR(50)
         COMMON/LTADM/LTADM(50)
         COMMON/LTPROD/LTPROD(50)
         COMMON/IRQTY/IRQTY(50)
         COMMON/IRL/IRL(50)
         COMMON/IROL/IROL(50)
         COMMON/ISUL/ISUL(50)
         COMMON/ITL/ITL(50)
         COMMON/RSIGLT/RSIGLT(50)
          COMMON/REQSIZ/REQSIZ(50)
         COMMON/UCOST/UCOST(50)
         DATA Z/1.0/
         DATA ITLMIN,IRLMIN/99999999,99999999/
   C
   C          COMPUTE PLANNING FACTORS
   C
          AMDA=COSHRT
         RLTM=LTADM(N)+LTPROD(N)
         IF(RLTM.LT.1.)RLTM=0.5
   ****
         AD=ADR(N)
         RLT=AD*RLTM/12.
         RMR=AD/12.
         UC=UCOST(N)
         ADDR=UC*AD
   C
   C          ESTIMATE STD DEVIATION OF FORECAST ERRORS
   C
         SIG=RSIGLT(N)
   C
   C     SET COR EQUAL TO SMALL PURCHASE ORDER COST
         COR=COSORD(1)
   C
   C          BRANCH BY EOQ FORMULA CODE
```

LEVELN,
p. 2 of 4

```
C
      GO TO (110,120,130,140,150,160,170,180,190),ICDEOQ
C
C        2 FEB 70 GUIDELINE LTR POLICY
C
  110 IF(ADDR.GE.1000.) GO TO 62
      Q=12.*RMR
      GO TO 200
   62 IF(ADDR.GE.5000.) GO TO 64
      Q=6.*RMR
      GO TO 200
   64 Q=3.*RMR
      GO TO 200
C
C                AFLC 57-6 POLICY---DEC 1976
C
  120 Q=SQRT(2.*COR*AD/(COSHLD*UC))
      ACOST=Q*UC
      IF(ACOST.LT.CSTBRK) GO TO 200
      COR=COSORD(2)
      Q=SQRT(2.*COR*AD/(COSHLD*UC))
      GO TO 200
  130 CONTINUE
  140 CONTINUE
  150 CONTINUE
  160 CONTINUE
  170 CONTINUE
  180 CONTINUE
  190 CONTINUE
      WRITE(6,8195)ICDEOQ,ICDSL
 8195 FORMAT('****LEVELN--UNDEFINED FORMULA CODE....',
     &        ' ICDEOQ=',I3,'  ICDSL=',I3)
      STOP
C
C        CHECK ORDER SIZE LIMITS
  200 CONTINUE
      EMX=EOQMAX*RMR
      IF(Q.GT.EMX) Q=EMX
      EMX=EOQMIN*RMR
      IF(Q.LT.EMX) Q=EMX
      IF(Q.LT. 1.)Q= 1.
C
C        BRANCH BY SAFETY LEVEL FORMULA CODE ICDSL
C
      GO TO (310,320,330,340,350,360,370,380,390),ICDSL
C
C        2 FEB 70 GUIDELINE LTR POLICY
C
  310 SL=1.*RMR
      GO TO 500
C
C        23 AUG 68 GUIDELINE LTR POLICY
```

LEVELN,
p. 3 of 4

```
1  04-16-77   12.082
      C
        320 SL=0.25*RLT
            GO TO 500
      C
      C        AFLCM 57-6 POLICY
      C
        330  Z=SQRT(REQSIZ(N))
             GO TO 358
      C
      C                PT-FORMULA TO MINIMIZE UNITS BACKORDERED
      C
        340  Z=1.
             GO TO 358
      C
      C                PT-FORMULA TO MINIMIZE REQUISITIONS BACKORDERED
      C
        350  Z=REQSIZ(N)
      C
      C
        358  CONTINUE
             RNUM=AMDA*SIG*(1.-EXP(-1.414*Q/SIG))
             DNOM=2.*COSHLD*UC*Z*1.414*Q
             RK=0.707*ALOG(RNUM/DNOM)
             SL=RK*SIG
             GO TO 500
      C
      C
        360  CONTINUE
        370  CONTINUE
        380  CONTINUE
        390  CONTINUE
             WRITE(6,8195)ICDEOQ,ICDSL
             STOP
      ***********************************************************************
      *****
        500  CONTINUE
      C
      C                LIMIT SAFETY LEVEL
      C                    FIRST,CHECK MINIMUM
             SLM=SLMIN*RMR
             IF(SL.LT.SLM)SL=SLM
      C
      C                LIMITS SL TO LESS THAN LEADTIME DEMAND
      C
             SLM=RLT*RMR
             IF(SL.GT.SLM)SL=SLM
      C
      C                LIMIT SL TO LESS THAN 3*SIG
      C
             SLM=3.*SIG
             IF(SL.GT.SLM)SL=SLM
      **
```

LEVELN (CONTINUE

p. 4 of 4

```
************************************************************
C
C       COMPUTE LEVELS
C
        IROTY(N)=Q+0.5
        IROL(N)=SL+RLT+0.5
        ITL(N)=SLMAX*RMR+RLT+GTLF*RMR+0.5
        IRL(N)=FLOAT(ITL(N))+GRLF*RMR+0.5
        IF(ITL(N).LE.ITLMIN) ITL(N)=ITLMIN
        IF(IRL(N).LE.IRLMIN)IRL(N)=IRLMIN
        ISUL(N)=GSULF*RLT
 3000   CONTINUE
        ROL=FLOAT(IROL(N))
         IF(IHBUG.NE.1) RETURN
         WRITE(6,8903)N,IROTY(N),IROL(N),ITL(N),IRL(N),ISUL(N)
 8903    FORMAT(4X,'****LEVELN--N=',I5,'  IROTY=',I5,'  IROL=',I5,
     &          '  ITL=',I5,'  IRL=',I5,'  ISUL=',I5)
         RETURN
        END
```

Subroutine: MAIN

Function:

MAIN is the primary INSSIM routine. This routine coordinates input of data, initialization of the simulation, event sequencing, and output reporting.

Description:

This routine is described in detail in Vol I, Chap V.

```
20      $      IDENT    WP0287,ADDSC-CALLAHAN
30      $      FORTY    DECK
40      $      OPTION   FORTRAN
50      $      PRMFL    C*,W,S,REQS/MAIN.0
60      $      LIMITS   5,32K
70      $      REMOTE   P*,AC
80      $      REMOTE   $$,AC
90             CHARACTER TEXT*40
100            INTEGER PX(52),PY(52,4)
110     *-------------------------------------------------------
120           COMMON/BPER/BPER
130           COMMON/COSHLD/COSHLD
140           COMMON/COSORD/COSORD(3)
150            COMMON/CSTBRK/CSTBRK
160           COMMON/COSHRT/COSHRT
170            COMMON/FSN/ALC,FSN(4),UM,NOUN(2),MGTCD(4),IOH,IOR,IPPL,IR
180           COMMON/GMRQF/GMRQF
190           COMMON/GRLF/GRLF
200           COMMON/GROQF/GROQF
210           COMMON/GSLF/GSLF
220           COMMON/GSULF/GSULF
230           COMMON/GTLF/GTLF
240           COMMON/GUIDI/ICDGID,NPH,NPOLCY,IDJAXR,IDJAXS,SHORT(10)
250           COMMON/GUIDII/DEMFAC,ALTMAX,PLTMAX,GRPSEL,IBEGIN,IEND
260           COMMON/ICDFOR/ICDFOR
270           COMMON/ICDSIG/ICDSIG
280           COMMON/IDBUG/IDBUG
290            COMMON/IEBUG/IEBUG
300           COMMON/IGBUG/IGBUG
310           COMMON/IFBUG/IFBUG
320            COMMON/IHBUG/IHBUG
330           COMMON/IDDIV/IDDIV
340            COMMON/IDENT/IDENT,IRUN,IREPL
350           COMMON/IDLEVL/IDLEVL
360           COMMON/IDSTAT/IDSTAT
370           COMMON/IDTHQ/IDTHQ
380           COMMON/IKDEM/IKDEM
390            COMMON/INLU/INLU
400            COMMON/INTYPE/INTYPE
410           COMMON/INQTR/INQTR
420           COMMON/IOBLIG/IOBLIG
430           COMMON/IPFOR/IPFOR
440           COMMON/IQTRND/IQTRND
450           COMMON/ISTAT/ISTAT
460           COMMON/ISTOCK/ISTOCK,IDSTOC
470           COMMON/ISTOP/ISTOP
480           COMMON/ITCANB/ITCANB,IDCANB
```

```
      COMMON/ITDAY/ITDAY
      COMMON/ITDIV/ITDIV
      COMMON/ITFOR/ITFOR,IDFOR
      COMMON/ITHQ/ITHQ
      COMMON/ITINV/ITINV
      COMMON/ITIME/ITIME
      COMMON/ITLEVL/ITLEVL
      COMMON/ITMNTH/ITMNTH
      COMMON/ITQTR/ITQTR
      COMMON/ITWEEK/ITWEEK
      COMMON/ITYEAR/ITYEAR
      COMMON/MAXBUY/MAXBUY
      COMMON/MBODAY/MBODAY
      COMMON/NBPER/NBPER
      COMMON/NBMAX/NBMAX
       COMMON/NDEM/NDEM
       COMMON/NDHIS/NDHIS
      COMMON/NENTRY/NENTRY
      COMMON/NFEMAX/NFEMAX
      COMMON/NFIRST/NFIRST
      COMMON/NITEM/NITEM
      COMMON/NLOC/NLOC
      COMMON/NLOCBK/NLOCBK
       COMMON/NRELP/NREPL
      COMMON/NTIME/NTIME
      COMMON/POLICY/ICDEOQ,ICDSL,EOQMAX,EOQMIN,SLMAX,SLMIN,RLF,
      COMMON/ADR/ADR(1)
      COMMON/IDEMND/IDEMND(1,24)
       COMMON/IRETUR/IRETUR(1,24)
       COMMON/IREQ/IREQ(1,24)
      COMMON/INVACT/INVACT(1)
      COMMON/INVDUE/INVDUE(1)
      COMMON/IRL/IRL(1)
      COMMON/IROL/IROL(1)
      COMMON/IRQTY/IRQTY(1)
      COMMON/ISUL/ISUL(1)
      COMMON/ITL/ITL(1)
      COMMON/LTADM/LTADM(1)
      COMMON/LTPROD/LTPROD(1)
      COMMON/XNT    /XNT(1)
      COMMON/NBOPT/NBOPT(1)
      COMMON/NBOTU/NBOTU(1)
      COMMON/NDEMAC/NDEMAC(1)
       COMMON/NRETAC/NRETAC(1)
       COMMON/NREQAC/NREQAC(1)
       COMMON/NDEMND/NDEMND(1,24)
       COMMON/NRETUR/NRETUR(1,24)
       COMMON/NREQ/NREQ(1,24)
      COMMON/NDENT/NDENT(1)
      COMMON/NORDPT/NORDPT(1)
      COMMON/REQMAD/REQMAD(1)
      COMMON/REQSIZ/REQSIZ(1)
      COMMON/RERSUM/RERSUM(1)
```

```
1020          COMMON/RMAD  /RMAD(1)
1030          COMMON/RMEAN/RMEAN(1)
1040          COMMON/RMREQS/RMREQS(1)
1050          COMMON/RMTBR /RMTBR(1)
1060          COMMON/RSIGLT/RSIGLT(1)
1070          COMMON/RTREND/RTREND(1)
1080          COMMON/UCOST/UCOST(1)
1090          COMMON/ILOCFE/ILOCFE(500)
1100          COMMON/JFSN/JFSN(500)
1110          COMMON/JPOINT/JPOINT(500)
1120          COMMON/JPRIOR/JPRIOR(500)
1130          COMMON/JQTY/JQTY(500)
1140          COMMON/JTIME/JTIME(500)
1150          COMMON/JTYPE/JTYPE(500)
1160          COMMON/IBACPT/IBACPT(100)
1170          COMMON/IDFSNB/IDFSNB(100)
1180          COMMON/ILOCBK/ILOCBK(100)
1190          COMMON/IPRIOR/IPRIOR(100)
1200          COMMON/IQTYB/IQTYB(100)
1210          COMMON/ITMBAC/ITMBAC(100)
1220          COMMON/IBOP/IBOPOH(3),IBOPOR(3)
1230          COMMON/IBAUTH/IBAUTH(12)
1240          COMMON/IBPROJ/IBPROJ(12)
1250          COMMON/GEOQ/GEOQ(3)
1260          COMMON/GEOQF/GEOQF(3)
1270          COMMON/IBACKI/IBACKI(30,3)
1280          COMMON/IBACKT/IBACKT(30,3)
1290          COMMON/IBAKDI/IBAKDI(30,3)
1300          COMMON/IBAKDT/IBAKDT(30,3)
1310          COMMON/ICANCL/ICANCL(30,3)
1320          COMMON/IDISPS/IDISPS(30,3)
1330          COMMON/IEXPED/IEXPED(30,3)
1340          COMMON/IFILLI/IFILLI(30,3)
1350          COMMON/IFILLT/IFILLT(30,3)
1360          COMMON/INVOH/INVOH(30,3)
1370          COMMON/INVDAY/INVDAY(30,3)
1380          COMMON/INVOR/INVOR(30,3)
1390          COMMON/IORDER/IORDER(30,3)
1400          COMMON/IRATON/IRATON(30,3)
1410          COMMON/IRECET/IRECET(30,3)
1420          COMMON/IREQC /IREQC(30,3)
1430          COMMON/IREQI /IREQI(30,3)
1440          COMMON/IREQT/IREQT(30,3)
1450          COMMON/IRETRN/IRETRN(30,3)
1460          COMMON/ISHIPI/ISHIPI(30,3)
1470          COMMON/ISHIPT/ISHIPT(30,3)
1480          COMMON/ITERM /ITERM(30,3)
1490           COMMON/ITRACE/ITRACE,ISTRAC
1500           DIMENSION CSHORT(4)
```

p. 4 of 9

```
*-------------------------------------------------------------
C
*          SET CODES FOR GUIDELINE REVISIONS
       NPH=20
       GTLF=24.
       GRLF=12.
       GSULF=0.
       GSLF=1.
       GROQF=1.
C
C
C                 READ INPUT DATA
C
  10     CONTINUE
         WRITE(6,8000)
 8000    FORMAT(1H1, T30, 'INVENTORY SYSTEM SIMULATOR'//T34,
&                 'RUN PARAMETERS'///)
C
C                 READ RUN IDENTIFICATION(CARD TYPE C1)
C
         READ(5,8001,END=9999)IDENT,TEXT
 8001    FORMAT(V)
         WRITE(6,8002)IDENT,TEXT
 8002    FORMAT(' (C1)  RUN-ID. ',I6,'  TITLE. ',A40)
C
C                 READ OUTPUT CONTROLS(CARD CODE C2)
C
         READ(5,8003)ITWRT,IOUT,IGRAPH,IPUNCH
 8003    FORMAT(V)
         WRITE(6,8004)ITWRT,IOUT,IGRAPH,IPUNCH
 8004    FORMAT(///' (C2)  OUTPUT CONTROLS...(NOTE. 1=YES)'//
&                  ' ITWRT ',T16,'IT.WRITE= ',I3//
&                  ' IOUT ',T16,'SUMMARY = ',I3//
&                  ' IGRAPH ',T16,'GRAPHS  = ',I3//
&                  ' IPUNCH ',T16,'PUNCH   = ',I3)
C
C                 READ DEBUG FLAGS
C
         READ(5,8003)IDBUG,IEBUG,IFBUG,IGBUG,IHBUG,ITRACE,ISTRAC
         WRITE(6,8005)IDBUG,IEBUG,IFBUG,IGBUG,IHBUG,ITRACE,ISTRAC
 8005    FORMAT(///' (C3)  DEBUG FLAGS'//
&                 T16,'IDBUG = ',I3//
&                 T16,'IEBUG = ',I3//
&                 T16,'IFBUG = ',I3//
&                 T16,'IGBUG = ',I3//
&                 T16,'IHBUG = ',I3///
&                  ' ITRACE ',T16,'START TRACE AT ',I6,'  FOR ITEM 1'/
&                  ' ISTRAC ',T16,'STOP TRACE AT ',I6)
C
C                 DEFINE ITEM INPUT FILES
C
         READ(5,8003)INLU,INTYPE,NDEM
         WRITE(6,8010)INLU,INTYPE,NDEM
 8010    FORMAT(///' (C4)  ITEM INPUT FILES'//
&                  ' INLU ',T16,'FILE = ',I3/
&                  ' INTYPE ',T16,'TYPE = ',I3,'  (1 = BCD; 2 = BINARY
&                  ' NDEM ',T16,'QTRS = ',I3)
C
```

p.5 of 9

```
C               READ MANAGEMENT METHOD CODES(CARD TYPE C5)
C
            READ(5,8003)ICDFOR,ICDSIG,ICDEOQ,ICDSL,ICDSLL,ICDBG,ICDSR
            WRITE(6,8050)ICDFOR,ICDSIG,ICDEOQ,ICDSL,ICDSLL,ICDBG,ICDB
       8050 FORMAT(1H1,/' (C5) MANAGEMENT METHODS TO BE USED'//
    &                ' ICDFOR',T16,'FORECAST FORMULA  =',I3//
    &                ' ICDSIG',T16,'STD DEV  FORMULA  =',I3//
    &                ' ICDEOQ',T16,'EOQ      FORMULA  =',I3//
    &                ' ICDSL ',T16,'SAFETY STK   CODE =',I3//
    &                ' ICDSLL',T16,'SAFETY LIMIT CODE =',I3//
    &                ' ICDBG ',T16,'BUDGET GUIDE CODE =',I3//
    &                ' ICDSR ',T16,'RETURNS CODE      =',I3)
C
C                READ MANAGEMENT PARAMETERS(CARD CODE C6)
C
            READ(5,8003)EOQMIN,EOQMAX,SLMIN,SLMAX
            WRITE(6,8060)EOQMIN,EOQMAX,SLMIN,SLMAX
       8060 FORMAT(///,' (C6) MANAGEMENT PARAMETERS'//
    &                ' EOQMIN',T16,'MIN EOQ(MNTHS)',T40,F10.4//
    &                ' EOQMAX',T16,'MAX EOQ(MNTHS)',T40,F10.4//
    &                ' SLMIN',T16,'MIN SAFETY LV(MNTHS)',T40,F10.4//
    &                ' SLMAX',T16,'MAX SAFETY LV(MNTHS)',T40,F10.4)
C
C               READ SYSTEM PARAMETERS(CARD TYPE C7)
C
            READ(5,8003)COSHLD,CSHORT,COSORD(1),COSORD(2),CSTBRK
            WRITE(6,8070)COSHLD,CSHORT,COSORD(1),COSORD(2),CSTBRK
       8070 FORMAT(///' (C7) SYSTEM PARAMETERS'///
    &                ' COSHLD',T16,'HOLDING COST/$-INV',T40,F10.4//
    &                ' CSHORT',T16,'SHORTAGE COST',T40,4F10.4///
    &               'COSORD(1)',T16,'SMALL ORDER COST',T40,F10.4//
    &               'COSORD(2)',T16,'LARGE ORDER COST',T40,F10.4///
    &               'CSTBRK',T16,'COST BREAK-POINT',T40,F10.4)
C
C               READ SIMULATION SIZE PARAMETERS(CARD TYPE C8)
C
            READ(5,8003)NRUN,INQTR,NREPL,NITEM,NDHIS
            WRITE(6,8080)NRUN,INQTR,NREPL,NITEM,NDHIS
       8080 FORMAT(///' (C8) SIMULATION SIZE'///
    &                ' NRUN',T16,'NUMBER OF RUNS',T40,I5//
    &                ' INQTR',T16,'NUMBER OF QUARTERS',T40,I5//
    &                ' NREPL',T16,'NO. OF REPLICATIONS',T40,I5//
    &                ' NITEM',T16,'NO. OF ITEMS/REPL',T40,I5//
    &                ' NDHIS',T16,'NO. OF HISTORY QTRS',T40,I5)
C
C               END OF RUN DEFINITION INPUT
C
C
```

p. 6 of 9

```
C*********************************************************
C
C              BEGIN SIMULATION
C
C*********************************************************
      DO 99 MRUN=1,NRUN
      IRUN=MRUN
      COSHRT=CSHORT(MRUN)
      WRITE(6,8090)COSHRT
 8090 FORMAT(1H1,25('****'),2(/'**'),
&              T20,'BEGIN SIMULATION WITH COSHRT =',F10.4,
&              2(/'**'),25('****'))
C
C              INITIALIZE RANDOM NUMBER STREAM
C
      R=RANDU(-.1)
C
C
      REWIND INLU
C
C              BEGIN REPLICATION LOOP
C
      CALL ZERO
      DO 95 KREPL=1,NREPL
      IPLOT=0
      IREPL=KREPL
   15 CONTINUE
C
C              INITIALIZE THE FUTURE EVENTS LIST(INITAL),
C                 AND READ ITEM DEMAND DATA (INITEM).
C
      CALL INITAL
      CALL INITEM
      IF(IDBUG .NE. 1) GO TO 16
      CALL WRIFEL
   16 CONTINUE
C
C******************************************
C
C
C        REMOVE NEXT EVENT FROM FUTURE EVENTS LIST
C      ***BRANCH BY TYPE OF TRANSACTION***
   20 CALL REMOVE(ITIME,KTYPE,IP3,IP4,IP5)
      GO TO ( 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 3
     C, 36, 37, 38, 39),KTYPE
C
C**********************************************
C
   21 CONTINUE
C        REFLECT RECEIPT OF TRANSACTION
C        REQUISITION
C
      CALL REQ(IP3,IP4,IP5,ITIME)
      GO TO 20
   22 CONTINUE
C        RECEIPT OF SHIPMENT
```

```
C        REFLECT RECEIPT
      CALL RECEIV(IP3,IP4)
      GO TO 20
   23 CONTINUE
C
C         REVIEW BACKORDERS.  CANCEL THOSE OLDER THAN MBODAY DAYS
C
      CALL CANCLB
      GO TO 20
C
C                SERVICEABLE RETURN EVENT
C
   24  CALL RET(IP3,IP4,ITIME)
       GO TO 20
   25 CONTINUE
C        STATUS REVIEW
      CALL STATUS
      GO TO 20
   26 CONTINUE
C        LEVELS COMPUTATION
      CALL LEVEL
      GO TO 20
   27 CONTINUE
C        BUY GUIDELINE
      CALL GUIDE
      GO TO 20
   28 CONTINUE
C        BUDGET REVIEW
      CALL BUDGET
      GO TO 20
   29 CONTINUE
C        FORECAST UPDATE
      CALL FORUPD
      GO TO 20
   30 CONTINUE
C***************************************
C
C         END OF RUN
C
C***************************************
       IF(ITWRT.EQ.1) CALL ITRSLT
       IF(KREPL.LT.NREPL) GO TO 95
  300 CONTINUE
       IF(IDBUG.EQ.1) CALL WRIFEL
      IF(IOUT .NE. 1) GO TO 90
      CALL OUT
       CALL OUTCST
   90 CONTINUE
       WRITE(6,302)
      IF(IGRAPH .EQ. 1) CALL PLOTR
      IF(IPUNCH.NE.1) GO TO 233
      DO 232 J=1,3
      DO 232 I=1,INQTR
```

p. 8 of 9

```
          WRITE(6,301)IDENT,MRUN,KREPL,J,I,
     &               IORDER(I,J),INVOH(I,J),INVOR(I,J),IBACKT(I,J),
     &               IBAKDT(I,J),IRETRN(I,J),IREQT(I,J),IFILLT(I,J)
  301   FORMAT(5I5,8I10)
  302   FORMAT(1H1,'IDENT MRUN  REP  J    QTR  ',
     &               T30,'IORDER ',T40,'INVOH ',T50,'INVOR ',T60,'IBACKT',
     &               T70,'IBAKDT ',T80,'IRETRN ',T90,'IREQT ',T100,'FILLT
  232 CONTINUE
  233 CONTINUE
      R=(ITIME-800)/ITYEAR+1
       NPLOT=1
      WRITE(6,8333)NPLOT,R
      WRITE(6,8330)
      WRITE(6,8332) (I,PX(I),(PY(I,J),J=1,4),I=1,IPLOT)
      GO TO 99
   31 CONTINUE
C        SPECIAL STATISTICS
      CALL SSTAT(IP3)
      GO TO 20
   32 CONTINUE
C        DEMAND PARAMETER UPDATE
      CALL DEMPAR(IP3,IP4,IP5)
      GO TO 20
   33 CONTINUE
       IF(ITIME.GT.ISTRAC) GO TO 332
C         RECORD HISTORY FOR ITEM NPLOT
      IPLOT=IPLOT+1
      PX(IPLOT)=ITIME
      PY(IPLOT,1)=INVACT(1)
      PY(IPLOT,2)=INVDUE(1)
      PY(IPLOT,3)=NBOTU(1)
      PY(IPLOT,4)=IOBLIG
       IF(IPLOT.LT.48) GO TO 331
      R=(ITIME-800)/ITYEAR+1
      WRITE(6,8333)NPLOT,R
      WRITE(6,8332) (I,PX(I),(PY(I,J),J=1,4),I=1,IPLOT)
 8330 FORMAT(//11X,4HWEEK,11X,4HTIME,8X,7HON-HAND,9X,6HDUE-IN,3[illegible]
     C-ORDERED  //)
 8332 FORMAT(52(6I15/))
 8333 FORMAT(1H1//20X,22HSTOCK HISTORY FOR ITEM,I5,10X,5HYEAR=,[illegible]
      IPLOT=0
C
```

```
C         WRITE BACKORDER FILE
C
      IF(IDBUG.NE.1) GO TO 331
      DO 298 N=1,NITEM
      IF(NBOPT(N).LE.0) GO TO 298
      JPT=NBOPT(N)
  288 CONTINUE
      I=JPT
      WRITE(6,8444)I,IDFSNB(I),IQTYB(I),IPRIOR(I),ITMBAC(I),IBA
 8444 FORMAT(I10,5I8)
      IF(IBACPT(JPT).EQ.0) GO TO 298
      JPT=IBACPT(JPT)
      GO TO 288
  298 CONTINUE
  331 CONTINUE
      INEXT=ITIME+ITWEEK
      CALL ENTER(INEXT,13,0,0,IPLOT)
      GO TO 20
C
C                  TURN OFF DEBUG SWITCH (IDBUG)
C
 332   IDBUG=0
       GO TO 20
   34 CONTINUE
   35 CONTINUE
   36 CONTINUE
   37 CONTINUE
   38 CONTINUE
   39 CONTINUE
C        KTYPE OUT OF RANGE-WRITE ERROR MESSAGE
      WRITE(6,113) KTYPE
  113 FORMAT(1H1,21H***ERROR***    KTYPE=,I3,3X,12HOUT OF RANGE)
       STOP
C
C                  END OF LOOP
C
  95   CONTINUE
   99 CONTINUE
      GO TO 10
 9999  CONTINUE
 9998  FORMAT(/////' ****MAIN--END OF FILE 05')
       STOP
      END
$      ENDJOB
C
```

Subroutine: ORDER

Function:

This routine performs the bookkeeping tasks required to reflect the placement of an order to replenish existing stocks, and to schedule the receipt of an order.

Description:

Subroutine ORDER first enters a type 2 event (receipt) onto the Future Events List. This will cause a receipt for IBQ units of item N a leadtime into the future. Next, ORDER updates the variables IOBLIG and INVDUE(N) which record the dollar values of obligations within the current budget period and the total number of units on order for item N, respectively. Performance variables IBOPSM and IBOPLG are next updated to reflect any small or large purchases which are placed in the initial quarter of the simulation. Finally, subroutine ORDER updates the pointer NORDPT(N) which points from item N to the corresponding receipt event on the Future Events List.

```
$     IDENT   WP0287,ADDSC-CALLAHAN
$     FORTY   DECK
$     PRMFL   C*,W,S,REQS/ORDER.O
$     REMOTE  P*,AC
$     REMOTE  $$,AC
     SUBROUTINE ORDER(N,IBQ,JTIME)
C      THIS ROUTINE UPDATES STATISTICS TO REFLECT AN ORDER FOR
C      UNITS OF ITEM N, WITH DELIVERY DATE SCHEDULED FOR JTIME
C      PLACES THE ASSOCIATED RECEIPT TRANSACTION ON THE FUTURE
C      LIST.
     COMMON/IDBUG/IDBUG
     COMMON/IOBLIG/IOBLIG
     COMMON/NLOC/NLOC
     COMMON/CSTBRK/CSTBRK
     COMMON/ISMORD/ISMORD(30,3)
     COMMON/ILGORD/ILGORD(30,3)
     COMMON/IORDER/IORDER(30,3)
     COMMON/NORDPT/NORDPT(50)
     COMMON/INVDUE/INVDUE(50)
     COMMON/UCOST/UCOST(50)
     COMMON/JPRIOR/JPRIOR(500)
      COMMON/ITIME/ITIME
      COMMON/IBOPCT/IBOPSM(3),IBOPLG(3)
C      PLACE ORDER FOR IBQ UNITS
     CALL ENTER(JTIME,2,N,IBQ,0)
     NPPT=NORDPT(N)
C      UPDATE STATUS STATISTICS
     IOBLIG=FLOAT(IOBLIG)+FLOAT(IBQ)*UCOST(N)
     INVDUE(N)=INVDUE(N)+IBQ
      DVORD=FLOAT(IBQ)*UCOST(N)
      IF(ITIME .LE. 0) GO TO 100
     CALL CUM(IORDER,IBQ,N)
     DVORD=FLOAT(IBQ)*UCOST(N)
     IF(DVORD .GE. CSTBRK) CALL CUM(ILGORD,IBQ,N)
     IF(DVORD .LT. CSTBRK) CALL CUM(ISMORD,IBQ,N)
      GO TO 200
 100  IDVORD=IFIX(DVORD)
      IF(DVORD .GE. CSTBRK) GO TO 150
      IBOPSM(1)=IBOPSM(1) + 1
      IBOPSM(2)=IBOPSM(2) + IBQ
      IBOPSM(3)=IBOPSM(3) + IDVORD
      GO TO 200
 150  IBOPLG(1)=IBOPLG(1) +1
      IBOPLG(2)=IBOPLG(2) + IBQ
      IBOPLG(3)=IBOPLG(3) + IDVORD
 200  CONTINUE
C     ARE ANY OTHER ORDERS OUTSTANDING ON THIS ITEM
     IF(NORDPT(N).GT.0) GO TO 11
     JPRIOR(NLOC)=0
     GO TO 12
C      UPDATE ORDER REFERENCE DATA
   11 JPRIOR(NLOC)=NORDPT(N)
C      SET ORDER POINTER
```

p. 2 of 2

```
   12 NORDPT(N)=NLOC
      IF(IDBUG.NE.1) GO TO 20
      WRITE(6,8000)N,IBQ,NORDPT(N),NPPT,NLOC
 8000 FORMAT(7H ORDER ,I26,7I8)
   20 CONTINUE
      RETURN
      END
$       ENDJOB
```

Subroutine: OUT

Function:

This routine produces a report of performance statistics which summarize results for the entire simulation run.

Description:

Subroutine OUT first updates the variables INVOH (I,1) and INVDAY (I,1) to record last quarter statistics, that is, results for period ITINV.

Subroutine OUT then produces 9 summary tables which describe the quarter by quarter activity observed during the simulation run. The tables appear in three separate formats, and each format is repeated with the performance statistics reported in terms of the number of actions that occurred, the number of units effected, and the total dollar value of these units, respectively.

At the bottom of each table the average per year for the simulation is printed. This value is obtained by totalling the quarter by quarter results within the same column, and dividing this total by the number of years that were simulated.

OUT

p. 1 of 4

```
$      IDENT   WPØ287,ADDSC-CALLAHAN
$      FORTY   DECK
$      PRMFL   C*,W,S,REQS/OUT.O
$      REMOTE  P*,AC
$      REMOTE  $$,AC
C              SUBROUTINE OUT
      SUBROUTINE OUT
      DIMENSION ITOTL(23,3),AVEYR(23)
       COMMON/INQTR/INQTR
      COMMON/ITIME/ITIME
      COMMON/ITYEAR/ITYEAR
      COMMON/ITINV/ITINV
      COMMON/INVACT/INVACT(5Ø)
      COMMON/NDENT/NDENT(5Ø)
       COMMON/NREPL/NREPL
      COMMON/NITEM/NITEM
      COMMON/IBOP/IBOPOH(3),IBOPOR(3)
      COMMON/IBACKI/IBACKI(3Ø,3)
      COMMON/IBACKT/IBACKT(3Ø,3)
      COMMON/IBAKDI/IBAKDI(3Ø,3)
      COMMON/IBAKDT/IBAKDT(3Ø,3)
      COMMON/ICANCL/ICANCL(3Ø,3)
      COMMON/IDISPS/IDISPS(3Ø,3)
      COMMON/IEXPED/IEXPED(3Ø,3)
      COMMON/IFILLI/IFILLI(3Ø,3)
      COMMON/IFILLT/IFILLT(3Ø,3)
      COMMON/IORDER/IORDER(3Ø,3)
      COMMON/INVDAY/INVDAY(3Ø,3)
      COMMON/INVOH/INVOH(3Ø,3)
      COMMON/INVOR/INVOR(3Ø,3)
      COMMON/IRATON/IRATON(3Ø,3)
      COMMON/IRECET/IRECET(3Ø,3)
      COMMON/IREQC /IREQC(3Ø,3)
      COMMON/IREQI /IREQI(3Ø,3)
      COMMON/IREQT/IREQT(3Ø,3)
      COMMON/IRETRN/IRETRN(3Ø,3)
      COMMON/ISHIPI/ISHIPI(3Ø,3)
      COMMON/ISHIPT/ISHIPT(3Ø,3)
      COMMON/ITERM /ITERM(3Ø,3)
C
C        TIDY UP LAST PERIOD STATISTICS
C
```

OUT (CONT)

p. 2 of 4

```
44Ø            I=ITINV
45Ø            INVDAY(I,1)=Ø
46Ø            INVOH(I,1)=Ø
47Ø            DO 5ØØ N=1,NITEM
48Ø            IF(NDENT(N).LT.Ø) GO TO 5ØØ
49Ø            INVOH(I,1)=INVOH(I,1)+1
5ØØ            IF(INVACT(N).LE.Ø) GO TO 5ØØ
51Ø            INVDAY(I,1)=INVDAY(I,1)+1
52Ø        5ØØ CONTINUE
53Ø            R=FLOAT(ITIME)/FLOAT(ITYEAR)
54Ø      C
55Ø             RN=R/FLOAT(NITEM)/FLOAT(NREPL)
56Ø            DO 2Ø1 I=1,23
57Ø            DO 2Ø1 J=1,3
58Ø            AVEYR(I)=Ø.
59Ø        2Ø1 ITOTL(I,J)=Ø
6ØØ            J=1
61Ø          2 CONTINUE
62Ø            ASSIGN 41 TO ISW
63Ø          1 IF(J.GT.3) GO TO 9ØØ
64Ø            WRITE(6,1ØØ)
65Ø            WRITE(6,1Ø2)
66Ø            WRITE(6,1Ø3)
67Ø            WRITE(6,1Ø2)
68Ø            GO TO(1Ø,2Ø,3Ø),J
69Ø         1Ø WRITE(6,1Ø4)
7ØØ            GO TO 4Ø
71Ø         2Ø WRITE(6,1Ø6)
72Ø            GO TO 4Ø
73Ø         3Ø WRITE(6,1Ø7)
74Ø         4Ø GO TO ISW,(41,51,52)
75Ø         41 WRITE(6,1Ø5)
76Ø            WPITE(6,1Ø8)
77Ø            WRITE(6,1Ø9)
78Ø            WRITE(6,11Ø)
79Ø            WRITE(6,121)IBOPOH(J),IBOPOR(J)
8ØØ             DO 5Ø I=1,INQTR
81Ø            ITOTL(1,J)=ITOTL(1,J)+INVOH(I,J)
82Ø            ITOTL(2,J)=ITOTL(2,J)+INVOR(I,J)
83Ø            ITOTL(3,J)=ITOTL(3,J)+IRECET(I,J)
84Ø            ITOTL(4,J)=ITOTL(4,J)+IRETRN(I,J)
85Ø            ITOTL(5,J)=ITOTL(5,J)+ISHIPT(I,J)
86Ø            ITOTL(6,J)=ITOTL(6,J)+ISHIPI(I,J)
87Ø            ITOTL(7,J)=ITOTL(7,J)+IORDER(I,J)
88Ø            ITOTL(8,J)=ITOTL(8,J)+IREQT(I,J)
89Ø            ITOTL(9,J)=ITOTL(9,J)+IREQC(I,J)
9ØØ            ITOTL(1Ø,J)=ITOTL(1Ø,J)+IREQI(I,J)
91Ø         5Ø WRITE(6,12Ø) I,INVOH(I,J),INVOR(I,J),IRECET(I,J),IRETRN(I
92Ø          CT(I,J),ISHIPI(I,J),IORDER(I,J),IREQT(I,J),IREQC(I,J),IREQ
93Ø            WRITE(6,11Ø)
```

OUT (CONT)

p. 3 of 4

```
940          WRITE(6,202) (ITOTL(I,J),I=1,1Ø)
950      2Ø2 FORMAT(1HØ,8H*TOTALS*,I1Ø,9(2X,I1Ø))
96Ø          DO 2Ø3 I=1,1Ø
97Ø      2Ø3 AVEYR(I)=FLOAT(ITOTL(I,J))/R
98Ø          WRITE(6,2Ø4) (AVEYR(I),I=1,1Ø)
99Ø      2Ø4 FORMAT(1HØ,6HAVE/YR,1X,1Ø(3X,F9.Ø))
1ØØØ          DO 3Ø5 I=1,1Ø
1Ø1Ø     3Ø5   AVEYR(I)=FLOAT(ITOTL(I,J))/RN
1Ø2Ø          WRITE(6,3Ø6)(AVEYR(I),I=1,1Ø)
1Ø3Ø     3Ø6   FORMAT(1HØ,6HAVE/IY,1X,1Ø(3X,F9.2))
1Ø4Ø         ASSIGN 51 TO ISW
1Ø5Ø         GO TO 1
1Ø6Ø       51  WRITE(6,111)
1Ø7Ø          WRITE(6,112)
1Ø8Ø         WRITE(6,67)
1Ø9Ø          DO 57 I=1,INQTR
11ØØ         ITOTL(11,J)=ITOTL(11,J)+IEXPED(I,J)
111Ø         ITOTL(12,J)=ITOTL(12,J)+IRATON(I,J)
112Ø         ITOTL(13,J)=ITOTL(13,J)+IDISPS(I,J)
113Ø         ITOTL(14,J)=ITOTL(14,J)+ITERM(I,J)
114Ø       57 WRITE(6,113) I,IEXPED(I,J),IRATON(I,J),IDISPS(I,J),ITERM(
115Ø          WRITE(6,67)
116Ø          WRITE(6,2Ø5)(ITOTL(I,J),I=11,14)
117Ø      2Ø5 FORMAT(1HØ,31X,8H*TOTALS*,4(2X,I1Ø))
118Ø         DO 2Ø6 I=11,14
119Ø      2Ø6 AVEYR(I)=FLOAT(ITOTL(I,J))/R
12ØØ          WRITE(6,2Ø7) (AVEYR(I),I=11,14)
121Ø      2Ø7 FORMAT(1HØ,33X,7HAVE/YR ,4(3X,F9.Ø))
122Ø          DO 315 I=11,14
123Ø     315   AVEYR(I)=FLOAT(ITOTL(I,J))/RN
124Ø          WRITE (6,316)(AVEYR(I),I=11,14)
125Ø     316   FORMAT(1HØ,33X,6HAVE/IY,4(3X,F9.2))
126Ø         ASSIGN 52 TO ISW
127Ø         GO TO 1
128Ø       52  WRITE(6,114)
129Ø          WRITE(6,115)
13ØØ          WRITE(6,116)
131Ø          WRITE(6,68)
132Ø          DO 58 I=1,INQTR
133Ø         IF(IREQT(I,J).LE.Ø) GO TO 62
134Ø         T1=FLOAT(IFILLT(I,J))/FLOAT(IREQT(I,J))
135Ø         GO TO 63
136Ø       62 T1=Ø.
137Ø       63 CONTINUE
138Ø         IF(IREQI(I,J).LE.Ø) GO TO 64
139Ø         T2=FLOAT(IFILLI(I,J))/FLOAT(IREQI(I,J))
14ØØ         GO TO 65
141Ø       64 CONTINUE
142Ø         T2=Ø.
143Ø       65 CONTINUE
144Ø         ITOTL(15,J)=ITOTL(15,J)+IBACKT(I,J)
145Ø         ITOTL(16,J)=ITOTL(16,J)+IBACKI(I,J)
```

OUT (CONT)

p. 4 of 4

```
      ITOTL(17,J)=ITOTL(17,J)+IBAKDT(I,J)
      ITOTL(18,J)=ITOTL(18,J)+IBAKDI(I,J)
      ITOTL(19,J)=ITOTL(19,J)+INVDAY(I,J)
      ITOTL(20,J)=ITOTL(20,J)+IFILLT(I,J)
      ITOTL(21,J)=ITOTL(21,J)+IFILLI(I,J)
   58 WRITE(6,117) I,IBACKT(I,J),IBACKI(I,J),IBAKDT(I,J),IBAKDI
     CINVDAY(I,J),IFILLT(I,J),IFILLI(I,J),T1,T2
      WRITE(6,68)
      WRITE(6,208) (ITOTL(I,J),I=15,21)
  203 FORMAT(1H0,8X,8H*TOTALS*,1X,7(I10,2X))
      DO 209 I=15,21
  209 AVEYR(I)=FLOAT(ITOTL(I,J))/R
      AVEYR(22)=FLOAT(ITOTL(20,J))/FLOAT(ITOTL(8,J))
      AVEYR(23)=FLOAT(ITOTL(21,J))/FLOAT(ITOTL(10,J))
  210 FORMAT(1H0,9X,7HAVE/YR   ,7F12.0,2F14.3)
      WRITE(6,210) (AVEYR(I),I=15,23)
       DO 325 I=15,23
 325   AVEYR(I)=FLOAT(ITOTL(I,J))/RN
       WRITE(6,329)(AVEYR(I),I=15,23)
 329   FORMAT(1H0,9X,6HAVE/IY,7F12.2,2F14.3)
      J=J+1
      GO TO 2
   67 FORMAT(1H ,35X,2H--,2X,4(12H    --------))
   68 FORMAT(1H ,11X,2H--,2X,7(12H    --------),2(14H    ------
  100 FORMAT(1H1,/,/,/,/)
  101 FORMAT(1H ,/,/)
  102 FORMAT(1H ,43X,44(1H-))
  103 FORMAT(1H ,43X,44HP E R F O R M A N C E   S T A T I S T
  104 FORMAT(1H ,/,/,54X,25H***   ACTIONS/FSN@S   ***,/)
  105 FORMAT(1H0,69X,10HPRIORITY 1)
  107 FORMAT(1H ,/,/,56X,19H$$$   DOLLARS   $$$)
  106 FORMAT(1H ,/,/,56X,17H...   UNITS    ...)
  108 FORMAT(1H ,10X,21HINVENTORY   INVENTORY,28X,67HTOTAL
     C    ORDERS       TOTAL          REGS        PRIORITY 1)
  109 FORMAT(1H ,128HPERIOD      ON HAND     ON ORDER     RECEIPTS
     CNS    SHIPMENTS   SHIPMENTS     PLACED   REQUISITIONS  CA
     CEQUISITIONS)
  110 FORMAT(1H ,2X,2H--,2X,10(4X,8(1H-)))
  111 FORMAT(1H0,54X,9HRATIONING)
  112 FORMAT(1H ,33X,57HPERIOD   EXPEDITES    ACTIONS      DISPO
     CMINATIONS)
  113 FORMAT(1H ,35X,I2,2X,4(4X,I8))
  114 FORMAT(1H0,44X,20HTOTAL      PRIORITY 1)
  115 FORMAT(1H ,20X,108HTOTAL      PRIORITY 1  BACKORDER   BACK
     CINVENTORY    TOTAL      PRIORITY 1   TOT FILLS      PRI 1 F
  116 FORMAT(1H ,9X,120HPERIOD   BACKORDERS BACKORDERS       DAYS
     CDAYS          DAYS         FILLS         FILLS        /TOT REG
     C1 REGS)
  117 FORMAT(1H ,11X,I2,2X,7(4X,I8),2(4X,F10.2))
  120 FORMAT(1H ,2X,I2,2X,10(4X,I8))
  121 FORMAT(1H ,1X,3HBOP,2X,2(4X,I8))
  999 RETURN
      END
 C     RETURN
```

Subroutine: OUTCST

Function:

This routine prints a summary of ordering activity observed during the simulation run.

Description:

This routine prints a summary table describing the number of order actions taken, the number of units effected, and the total dollar value of these orders. The results are segregated in terms of "small ordering actions" and "large ordering actions". A small ordering action is defined as an order whose total dollar value is less than CSTBRK dollars.

Finally, subroutine OUTCST prints averages per year of each of the statistics presented.

OUTCST
p. 1 of 2

```
11  04-16-77    12.087

      SUBROUTINE OUTCST
       COMMON/NITEM/NITEM
       COMMON/NREPL/NREPL
       COMMON/INOTR/INOTR
      COMMON/ISMORD/ISMORD(30,3)
      COMMON/ILGORD/ILGORD(30,3)
       COMMON/COSORD/COSORD(3)
      COMMON/IBOPCT/IBOPSM(3),IBOPLG(3)
      COMMON/CSTBRK/CSTBRK
      DIMENSION IDUM(3),ITOT(9)
      LW=6
       CORDSM=COSORD(1)
       CORDLG=COSORD(2)
      DO 1 I=1,9
    1 ITOT(I)=0
      WRITE(LW,100)
      WRITE(LW,200)
      WRITE(LW,800) CORDSM,CORDLG
      WRITE(LW,300)
      IDUM(1)=IBOPSM(1) + IBOPLG(1)
      IDUM(2)=IBOPSM(2) + IBOPLG(2)
      IDUM(3)=IBOPSM(3) + IBOPLG(3)
      WRITE(LW,700) IBOPSM,IBOPLG,IDUM
       DO 2 I=1,INOTR
      DO 3 J=1,3
      K=J + 3
      L=J + 6
      IDUM(J)=ISMORD(I,J) + ILGORD(I,J)
      ITOT(J)=ITOT(J) + ISMORD(I,J)
      ITOT(K)=ITOT(K) + ILGORD(I,J)
    3 ITOT(L)=ITOT(L) + IDUM(J)
    2 WRITE(LW,400) I,(ISMORD(I,J),J=1,3),(ILGORD(I,K),K=1,3),IDUM
      WRITE(LW,500) ITOT
      TOTSM=FLOAT(ITOT(1))*CORDSM
      TOTLG=FLOAT(ITOT(4))*CORDLG
      TOT=TOTSM + TOTLG
      WRITE(LW,600) TOTSM,TOTLG,TOT
      WRITE(LW,900) CSTBRK
       RN=FLOAT(NITEM*NREPL)
       TOTSM=TOTSM/RN
       TOTLG=TOTLG/RN
       TOT=TOTSM+TOTLG
       WRITE(LW,605)
605    FORMAT(//T10,'AVERAGE COST/ITEM/YR')
       WRITE(6,600)TOTSM,TOTLG,TOT
  100 FORMAT(1H1,48X,'*****      ORDER COST STATISTICS      *****')
  200 FORMAT(1H0,'TIME',10X,'SMALL ORDERS',18X,'LARGE ORDERS',18X,'TOTAL
     1 ORDERS')
  300 FORMAT(1H0,3X,3(4X,'NUMBER      UNITS    DOLLARS'))
  400 FORMAT(1H ,I3,4X,3(I6,I10,I10,4X))
  500 FORMAT(1H0,'TOTALS',1X,3(I6,I10,I10,4X))
  600 FORMAT(1H0,'SMALL ORDER COST ',E15.8,' PLUS LARGE ORDER COST ',E15
```

OUTCST
p. 2

```
     1,8,' EQUALS TOTAL ORDER COST ',E15.8)
 700 FORMAT(1H0,'BOP',4X,3(I6,I10,I10,4X))
 800 FORMAT(1H0,'COST TO ORDER',4X,F6.2,24X,F6.2)
 900 FORMAT(1H0,'EOQ BUY DOLLAR BREAK POINT ',F10.2)
     RETURN
     END
```

Subroutine: PLOTR

Function:

This routine plots peformance statistics verses time for several performance measures.

Description:

This routine calls subroutine GP to produce four separate types of plots. These are:

a. Requisitions, cancellations, and disposals by quarter.

b. On-hand, on-order, and backordered stock by quarter. Aggregate pipeline stock is also plotted, where aggregate stock equals on-hand plus on-order stocks.

c. Requisitions, fills, backorders, and cancellations by quarter.

d. Cumulative budgeted expenditures and cumulative obligations (i.e., dollar value of orders placed) by quarter.

PLOTR,
p. 1 of 2

```
02-11-77    10.691

        SUBROUTINE PLOTR
        COMMON /INVOH/INVOH(30,3)
        COMMON /INVOR/INVOR(30,3)
        COMMON /IBACKT/IBACKT(30,3)
        COMMON /ICANCL/ICANCL(30,3)
        COMMON/IDISPS/IDISPS(30,3)
        COMMON /IREQT/IREQT(30,3)
        COMMON /IFILLT/IFILLT(30,3)
        COMMON /IREQC/IREQC(30,3)
        COMMON /IRAUTH/IRAUTH(12)
        COMMON /IORDER/IORDER(30,3)
        DIMENSION X(31),Y(31,4),PLT(50,100),A(4)
        DIMENSION RMEAS(3)
        DATA (RMEAS(I),I=1,3)/6HFSNS  ,6HUNITS ,6HDOLLRS/
        DATA CH/1HH/,CO/1HO/,CB/1HB/,CA/1HA/,CR/1HR/,CE/1HE/,CC/1HC/
        DATA CD/1HD/
        X(31)=0.
        DO 5 I=1,4
  5     Y(31,I)=0.
        A(1)=CR
        A(2)=CC
        A(3)=CD
        DO 110 J=1,3
        WRITE(6,3)
        WRITE(6,8000) RMEAS(J)
        WRITE(6,8040)
 8040   FORMAT(10X,16HREQUISITIONS = R,5X,17HCANCELLATIONS = C,5X,13HDISPO
       1SALS = D)
        DO 105 I=1,30
        X(I)=FLOAT(I)
        Y(I,1)=FLOAT(IREQT(I,J))
        Y(I,2)=FLOAT(ICANCL(I,J))
        Y(I,3)=FLOAT(IDISPS(I,J))
  105   CONTINUE
        CALL GP(X,Y,0,16,31,3,100,40,A,PLT)
  110   CONTINUE
        A(1)=CH
        A(2)=CO
        A(3)=CB
        A(4)=CA
        DO 10 J=1,3
        DO 20 I=1,30
        Y(I,1)=FLOAT(INVOH(I,J))
        Y(I,2)=FLOAT(INVOR(I,J))
        Y(I,3)=FLOAT(IBACKT(I,J))
  20    Y(I,4)=FLOAT(INVOH(I,J))+FLOAT(INVOR(I,J))
        WRITE (6,3)
        WRITE(6,8010)
        WRITE(6,8000)RMEAS(J)
 8010   FORMAT(10X,11HON-HAND = H,5X,12HON-ORDER = O,5X,14HBACKORDERS = B,
       15X,13HAGGREGATE = A)
   10   CALL GP(X,Y,0,16,31,4,100,40,A,PLT)
```

PLOTR,
p. 2

```
02-11-77    10.691
      A(1)=CR
      A(2)=CF
      A(3)=CB
      A(4)=CC
      DO 30 J=1,3
      Y(1,1)=FLOAT(IREQT(1,J))
      Y(1,2)=FLOAT(IFILLT(1,J))
      Y(1,3)=FLOAT(IBACKT(1,J))
      Y(1,4)=FLOAT(ICANCL(1,J))
      DO 40 I=2,30
      Y(I,1)=FLOAT(IREQT(I,J))+Y(I-1,1)
      Y(I,2)=FLOAT(IFILLT(I,J))+Y(I-1,2)
      Y(I,3)=FLOAT(IBACKT(I,J))+Y(I-1,3)
   40 Y(I,4)=FLOAT(ICANCL(I,J))+Y(I-1,4)
      WRITE (6,3)
      WRITE(6,8020)
      WRITE(6,8000)RMEAS(J)
 8020 FORMAT(10X,16HREQUISITIONS = R,5X,9HFILLS = F,5X,14HBACKORDERS = B
     1,5X,17HCANCELLATIONS = C)
   30 CALL GP(X,Y,0,16,31,4,100,40,A,PLT)
      A(1)=CB
      A(2)=CO
      J=3
      Y(1,1)=FLOAT(IBAUTH(1))
      Y(1,2)=FLOAT(IORDER(1,J))
      DO 60 I=2,30
      KK=((I-1)/4)+1
      IBI=0
      DO 70 K=1,KK
 70     IBI=IBI+IBAUTH(K)
      Y(I,1)=FLOAT(IBI)
 60     Y(I,2)=FLOAT(IORDER(I,J))+Y(I-1,2)
      WRITE (6,3)
      WRITE(6,8030)
      WRITE(6,8000)RMEAS(J)
 8030 FORMAT(10X,10HBUDGET = B,5X,10HORDERS = O)
   50 CALL GP(X,Y,0,16,31,2,100,40,A,PLT)
    3 FORMAT('1')
 8000 FORMAT('0','MEASURED IN ...',A6)
       RETURN
       END
```

Subroutine: RANDU

Function:

This routine generates uniformally distributed random numbers on the interval 0. to 1.0.

Description:

This routine uses the congruential method to generate pseudo-random numbers. A call of the form

```
CALL RANDU (X)
```

where X is negative, causes the random number seed to be set to the absolute value of X. If X is positive, X has no effect on the calculation.

RANDU
p. 1 of 1

```
C                   RANDU--U(0,1) RANDOM NUMBER GENERATOR.
      FUNCTION RANDU(X)
C                   A CALL WITH X < 0. INITIALIZES THE RANDOM NUMBER STREAM.
      IF(X) 10,20,20
   20 RN=RHO*RANDU
      RN1=AMOD(RN,BN)
      RANDU=RN1/BN
      RETURN
   10 RHO=7.0**13
      BN=10.0**10
      RANDU=-X
      GO TO 20
      END
```

Subroutine: RECEIV

Function:

This routine updates stock status records to reflect the receipt of a replenishment order (event type 2) from a supplier of inventory system.

Description:

This routine updates stock status records to reflect the receipt of an order for IQTY units of item N during the current period. The routine then initiates shipment actions to fill any outstanding backorders.

By convention, order histories are maintained for each item in the form of a string-list ordered from the newest to the oldest outstanding order. When an outstanding order is received, this string must be updated to drop the last element on the string (orders are assumed not to cross). The variable NORDPT (N) indicates the position in the Future Events List of the most recent order for item N. Beginning with this record, the routine searches down the order string to locate the oldest order. The last record on this string is the oldest order placed.

By convention, orders are terminated by setting the order quantity to zero and removing the order transaction from the Future Events List at the regular due-in time.

1 04-16-77 12.090

RECEIV, p. 1 of 2

```
      SUBROUTINE RECEIV(N,IQTY)
C       THIS ROUTINE UPDATES STOCK STATUS RECORDS TO REFLECT RECEIPT
C       OF AN ORDER FOR IQTY UNITS OF ITEM N DURING PERIOD I.
C       THE ROUTINE THEN INITIATES SHIPMENT ACTIONS TO FULFILL
C       OUTSTANDING BACKORDERS IF ANY.
      COMMON/IDBUG/IDBUG
      COMMON/NLOCBK/NLOCBK
      COMMON/IRECET/IRECET(30,3)
      COMMON/ISHIPI/ISHIPI(30,3)
      COMMON/ISHIPT/ISHIPT(30,3)
      COMMON/JPRIOR/JPRIOR(500)
      COMMON/IQTYB/IQTYB(200)
      COMMON/IPRIOR/IPRIOR(200)
      COMMON/NBOTU/NBOTU(50)
      COMMON/ILOCBK/ILOCBK(200)
      COMMON/NORDPT/NORDPT(50)
      COMMON/INVACT/INVACT(50)
      COMMON/INVDUE/INVDUE(50)
      COMMON/NBOPT/NBOPT(50)
      COMMON/ISUL/ISUL(50)
      COMMON/IBACPT/IBACPT(200)
C       ---NOTE---
C       BY CONVENTION ORDER HISTORIES ARE MAINTAINED FOR EACH ITEM IN
C       THE FORM OF A STRING LIST ORDERED FROM THIS NEWEST TO THE
C       OLDEST OUTSTANDING ORDER. WHEN AN OUTSTANDING ORDER IS RECEIVED,
C       THIS STRING MUST BE UPDATED TO DROP THE LAST ELEMENT ON THE
C       STRING(ORDERS ARE ASSUMED NOT TO CROSS), THE NEXT FEW STATEMENT
C       ACCOMPLISH THIS UPDATE.
      NPT=NORDPT(N)
C       IS THIS THE ONLY OUTSTANDING ORDER FOR THIS ITEM
      IF(JPRIOR(NPT).NE.0) GO TO 10
C       SET POINTS TO INDICATE THERE IS NO LONGER AN OUTSTANDING ORDER
      NORDPT(N)=0
      GO TO 15
C       SET POINTERS TO SEARCH DOWN ORDER STRING
   10 NPP=NPT
      NPT=JPRIOR(NPP)
C       IS NPT THE OLDEST OUTSTANDING ORDER
      IF(IDBUG.NE.1) GO TO 12
      WRITE(6,991)NPP,NPT
   12 CONTINUE
  991 FORMAT(1H ,13HLOOP ENTERED-,7HJPRIOR=,I4,4HNPT=,I2)
      IF(JPRIOR(NPT).NE.0) GO TO 10
C       THEN NPT IS THE ORDER JUST RECEIVED, REMOVE THE RECORD FROM THE
C       END OF THE ORDER STRING
      JPRIOR(NPP)=0
C       BY CONVENTION ORDERS ARE TERMINATED BY SETTING THE ORDER
C       QUANTITY TO ZERO AND REMOVING THE ORDER TRANSACTION FROM THE
C       FUTURE EVENTS LIST AT THE REGULAR DUE-IN TIME.
C
C       HAS THIS ORDER BEEN TERMINATED.
   15 IF(IQTY.EQ.0) RETURN
```

RECEIV, p. 2 of 2

```
C               ADJUST GROSS RECEIPT STATISTICS
      JQTY=-IQTY
      CALL CUM(IRECET,IQTY,N)
C       ADJUST ASSETS FOR THIS ITEM
      INVACT(N)=INVACT(N)+IQTY
      INVDUE(N)=INVDUE(N)-IQTY
C       ARE THERE ANY BACKORDERS ON THIS ITEM.
      IF(NBOPT(N).LE.0) RETURN
      CALL FILLBO(N)
      RETURN
      END
```

Subroutine: REMOVE

Function:

This routine removes the earliest transaction from the Future Events List, and updates the chain list structure.

Description:

This routine determines the "current" event to be simulated within INSSIM. For a detailed description of the event bookkeeping process, see Volume I, Chapter IV.

REMOVE
p. 1 of 1

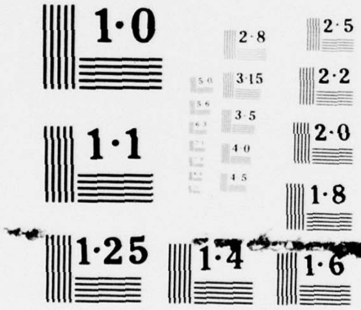
```
02-11-77    19.596

      SUBROUTINE REMOVE(KTIME,KTYPE,KESN,KQTY,KPRIOR)
C         THIS ROUTINE REMOVES THE EARLIEST TRANSACTION FROM THE
C         CURRENT EVENTS CHAIN.AND UPDATES THE CHAIN STRUCTURE
      COMMON/IDBUG/IDBUG
      COMMON/NENTRY/NENTRY
      COMMON/NFIRST/NFIRST
      COMMON/NTIME/NTIME
      COMMON/ILOCEE/ILOCEE(500)
      COMMON/JESN/JESN(500)
      COMMON/JPOINT/JPOINT(500)
      COMMON/JPRIOR/JPRIOR(500)
      COMMON/JQTY/JQTY(500)
      COMMON/JTIME/JTIME(500)
      COMMON/JTYPE/JTYPE(500)
      K=NFIRST
      KTIME=JTIME(K)
      KTYPE=JTYPE(K)
      KESN=JESN(K)
      KQTY=JQTY(K)
      KPRIOR=JPRIOR(K)
      NFIRST=JPOINT(K)
      NTIME=JTIME(NFIRST)
      IF(IDBUG.NE.1) GO TO 25
      WRITE(6,8000)KTIME,KTYPE,KESN,KQTY,KPRIOR,K
 8000 FORMAT(7H REMOVE,I10,5I8)
   25 CONTINUE
C         IS THIS THE LAST TRANSACTION ON THE LIST
      IF(NENTRY.GT.1) GO TO 19
C         SET LIST CLOCK UP A LARGE INCREMENT
      NTIME=9999999
C         RECORD LOCATION K AS AVAILABLE
   19 ILOCEE(NENTRY)=K
C         DECREMENT COUNT OF LIST ENTRIES
      NENTRY=NENTRY-1
      RETURN
      END
```

AD-A046 053   DECISION SYSTEMS DAYTON OH   F/G 15/5
THE INVENTORY SYSTEM SIMULATOR (INSSIM). VOLUME II PROGRAM LIST--ETC(U)
AUG 77 W S DEMMY   F49620-77-C-0063
UNCLASSIFIED   RM-77-03   AFOSR-TR-77-1235   NL

2 OF 2
ADA
046053

END
DATE
FILMED
12-77
DDC


1·0
2·8
2·5
5·0
3·15
2·2
5·6
6·3
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

Subroutine: REQ

Function:

This routine reflects the receipt of a customer requisition (event type 1), and associated shipping and/or backorder actions.

Description: This routine is called a statement of the form

CALL REQ (N, IQTY, IPRI, LTIME)

Where

N = item number associated with the current requisition event

IQTY = quantity associated with this requisition

IPRI = the priority of the requisition, where IPRI = 1 denotes a high priority requisition, and IPRI = 2 denotes a low priority requisition

LTIME = the time of receipt of the current requisition

First, the subroutine REQ updates the accumulators NDEMAC (N) and NREQAC (N). Next, the routine attempts to ship on hand assets to fill this requisition. For low priority requisitions (i.e. requisitions with a priority code = 2) units are shipped until on hand stock reaches the support level for the item. For high priority requisitions, shipments are made until on hand stock is reduced to zero.

If there is insufficient stock to completely satisfy the requisition, subroutine ENTERB is called to record a backorder for this item.

At the conclusion of the routine, subroutine CUM is called to update shipping and fillrate statistics.

REQ, p.1 of 2

```
      SUBROUTINE REQ(N,IQTY,IPRI,LTIME)
C       THIS ROUTINE REFLECTS PROCESSING OF A REQUISITION FOR IQTY
C       UNITS OF ITEM N DURING PERIOD I,WHERE IPRI=1 DENOTES A HIGH
C       PRIORITY REQUISITION, IPRI=2 DENOTES A LOW PRIORITY REQUISITION
C       JTIME DENOTES THE CLOCK TIME THE REQ WAS RECEIVED
C       RECORD REQUISITION DATA
      COMMON/REQSIZ/REQSIZ(50)
      COMMON/REQMAD/REQMAD(50)
      COMMON/NDEMAC/NDEMAC(50)
       COMMON/NREQAC/NREQAC(50)
      COMMON/IREQI /IREQI(30,3)
      COMMON/IREQT/IREQT(30,3)
      COMMON/INVACT/INVACT(50)
      COMMON/ISUL/ISUL(50)
      COMMON/ISHIPI/ISHIPI(30,3)
      COMMON/ISHIPT/ISHIPT(30,3)
      COMMON/IFILLT/IFILLT(30,3)
      COMMON/IFILLI/IFILLI(30,3)
      COMMON/NBOTU/NBOTU(50)
      CALL CUM(IREQT,IQTY,N)
C       IS THIS A PRIORITY 1 REQ
      IF(IPRI.NE.1) GO TO 20
      CALL CUM(IREQI,IQTY,N)
C       UPDATE DEMAND ACCUMULATOR AND EXPONENTIALLY SMOOTHED
   20 NDEMAC(N)=NDEMAC(N)+IQTY
       NREQAC(N)=NREQAC(N)+1
C       IS THERE ANY STOCK ON HAND
      IF(INVACT(N).GT.0) GO TO 40
C       PUT THIS REQUISITION IN BACKORDER STATUS
      CALL ENTERB(N,IQTY,IPRI,LTIME)
C       UPDATE BACKORDER STATISTICS
      NBOTU(N)=NBOTU(N)+IQTY
      RETURN
C       SET ITEST=STOCK REMAINING IF THE ORDER WERE FILLED COMPLETELY
   40 ITEST=INVACT(N)-IQTY
C       IS ITEST ABOVE THE SUPPORT LEVEL
      IF(ITEST.GT.ISUL(N)) GO TO 80
C       IS THIS REQ PRIORITY 1
      IF(IPRI.NE.1) GO TO 60
C       CAN THE REQUISITION BE COMPLETELY SATISFIED FROM STOCK ON-HAND
      IF(INVACT(N).GE.IQTY) GO TO 80
C       SHIP ALL REMAINING ON-HAND STOCK IN PARTIAL FULFILLMENT
C       OF THIS REQUISITION
      IQTYS=INVACT(N)
C       COMPUTE QUANTITY TO BE BACKORDERED AND UPDATE PRI-1 STATS
      IBK=IQTY-IQTYS
      CALL CUM(ISHIPI,IQTYS,N)
      CALL CUMB(IFILLI,IQTYS,N)
      GO TO 65
C       ARE ON-HAND ASSETS ABOVE THE SUPPORT LEVEL
   60 IF(INVACT(N).LE.ISUL(N)) GO TO 70
C       SHIP DOWN TO THE SUPPORT LEVEL
```

REQ (CONTINUED)
p 2 of 2

```
      IQTYS=INVACT(N)-ISUL(N)
C       BACKORDER THE REMAINING QUANTITY
      IBK=IQTY-IQTYS
   65 CALL ENTERB(N,IBK,IPRI,LTIME)
C       UPDATE STOCK STATUS RECORDS
      INVACT(N)=INVACT(N)-IQTYS
      NBOTU(N)=NBOTU(N)+IBK
C       UPDATE GROSS PERFORMANCE STATISTICS
      JQTYS=-IQTYS
      CALL CUM(ISHIPT,IQTYS,N)
      CALL CUMB(IFILLT,IQTYS,N)
      RETURN
   70 CONTINUE
C         BACKORDER ENTIRE REQUISITION
      CALL ENTERB(N,IQTY,IPRI,LTIME)
C         UPDATE STORK STATUS RECORDS
C         UPDATE STOCK STATUS RECORDS
      NBOTU(N)=NBOTU(N)+IQTY
      RETURN
C       SHIP TO FILL THE ENTIRE REQUISITION
   80 IQTYS=IQTY
C       UPDATE THE STOCK STATUS RECORDS
      INVACT(N)=INVACT(N)-IQTYS
C       UPDATE PERFORMANCE
      JQTYS=-IQTYS
      CALL CUM(ISHIPT,IQTYS,N)
      CALL CUM(IFILLT,IQTYS,N)
C       IS THIS A PRIORITY 1 REQUISITION
      IF(IPRI.NE.1) GO TO 85
C       UPDATE PRIORITY 1 STATISTICS
      CALL CUM(ISHIPI,IQTYS,N)
      CALL CUM(IFILLI,IQTYS,N)
   85 CONTINUE
      RETURN
      END
```

Subroutine: RET

Function:

This routine reflects the receipt of serviceable returns by the supply system (event code = 4).

Description:

First, the accumulator NRETAC (N) is updated to reflect the receipt of IQTY units of item N. Next, the performance statistic IRETRN is updated, and onhand assets INVACT (N) are increased. Finally, subroutine FILLBO is called in an attempt to fill any outstanding backorders.

RET
p. 1 of 1

```
)1  04-16-77    12.093
        SUBROUTINE RET(N,IQTY,ITIME)
        COMMON/NRETAC/NRETAC(1)
        COMMON/IRETRN/IRETRN(30,3)
        COMMON/INVACT/INVACT(1)
        COMMON/NBOPT/NBOPT(1)
*
*        UPDATE SERVICEABLE RETURN ACCUMULATOR
*
        NRETAC(N)=NRETAC(N) + IQTY
*
*        UPDATE AGGREGATE PERFORMANCE STATISTICS
*
        CALL CUM(IRETRN,IQTY,N)
*
*        UPDATE CURRENT ON-HAND ASSETS FOR ITEM N
*
        INVACT(N)=INVACT(N) + IQTY
*
*        ARE THERE ANY BACKORDERS ON THIS ITEM
*
        IF(NBOPT(N) .LE. 0) RETURN
*
*        ELSE REDUCE BACKORDERS BY AMOUNT OF RETURNS
*
        CALL FILLBO(N)
        RETURN
        END
```

Subroutine: SSTAT

Function:

This routine updates time persistent and end of period statistics.

Description:

This routine is called at the end of each simulation week. Subroutine SSTAT first updates the performance statistics INVDAY, IBAKDT, and IBAKDI. These variables record inventory-weeks and backorder week measures.

At the end of each quarter, the routine computes values for the end of period variables INVOH, INVOR, IVACKT, and IBACKI. These variables record the end of period status for on hand stocks, on order stocks, and total backorders and priority 1 backorders. The period counters KEND (which denotes the week number within the quarter) and ITINV (the current quarter number) are also incremented by one at the end of each quarter.

Finally, this routine ends by entering a new type 11 event onto the Future Events List.

SSTAT

p 1 of 2

```
$       IDENT   WP0287,ADDSC-CALLAHAN
 $       FORTY   DECK
$       PRMFL   C*,W,S,REQS/SSTAT.O
$       REMOTE  P*,AC
$       REMOTE  $$,AC
       SUBROUTINE SSTAT(KEND)
C                SUBROUTINE SSTAT
       COMMON/IDSTAT/IDSTAT
       COMMON/ISTAT/ISTAT
       COMMON/ISTOP/ISTOP
       COMMON/ITIME/ITIME
       COMMON/ITINV/ITINV
       COMMON/ITQTR/ITQTR
       COMMON/NITEM/NITEM
        COMMON/NBMAX/NBMAX
        COMMON/NLOCBK/NLOCBK
       COMMON/INVACT/INVACT(50)
       COMMON/INVDUE/INVDUE(50)
       COMMON/NDENT/NDENT(50)
       COMMON/IBACKI/IBACKI(30,3)
       COMMON/IBACKT/IBACKT(30,3)
       COMMON/IBAKDT/IBAKDT(30,3)
       COMMON/IBAKDI/IBAKDI(30,3)
       COMMON/INVDAY/INVDAY(30,3)
       COMMON/INVOH/INVOH(30,3)
       COMMON/INVOR/INVOR(30,3)
       COMMON/NBOTU/NBOTU(50)
       COMMON/UCOST/UCOST(50)
       I = ITINV
       DO 50 N= 1,NITEM
       IF(NDENT(N).LE.0) GO TO 50
       IF(INVACT(N).LE.0)GO TO 20
       INVDAY(I,1) = INVDAY(I,1) + 1
       INVDAY(I,2) =INVDAY(I,2) + INVACT(N)
       INVDAY(I,3) = INVDAY(I,3) + IFIX(UCOST(N)*FLOAT(INVACT(N))
        GO TO 50
    20 CONTINUE
       IF(NBOTU(N).LE.0)GO TO 50
        NRS=NBOTU(N)
        CALL CUMB(IBAKDT,NRS,N)
        CALL CUMB(IBAKDI,NRS,N)
    50 CONTINUE
C
```

SSTAT (CONT)

p. 2 of 2

```
C                   UPDATE REQUISITIONS-SHORT COUNT
C
        NRS=NBMAX-NLOCBK
        IF(NRS.LE.0) GO TO 60
        IBAKDT(I,1)=IBAKDT(I,1)+NRS
        IBAKDI(I,1)=IBAKDI(I,1)+NRS
   60   CONTINUE
C
       KP = ITQTR/IDSTAT
       IF(KEND.GE.KP)GO TO 100
       KEND = KEND + 1
       GO TO 200
C
C                   UPDATE END OF PERIOD STATISTICS
C
  100   CONTINUE
       DO 150 N=1,NITEM
       IF(NDENT(N).LE.0) GO TO 150
       IF(INVACT(N).GT.0) CALL CUM(INVOH,INVACT(N),N)
       IF(NBOTU(N).GT.0) CALL CUMB(IBACKT,NBOTU(N),N)
        IF(NBOTU(N).GT.0) CALL CUMB(IBACKI,NBOTU(N),N)
       IF(INVDUE(N).GT.0) CALL CUM(INVOR,INVDUE(N),N)
  150  CONTINUE
C
C                   UPDATE COUNT OF EOP REQUISITIONS SHORT
C
        NRS=NBMAX-NLOCBK
        IF(NRS.LE.0) GO TO 160
        IBACKT(I,1)=IBACKT(I,1)+NRS
        IBACKI(I,1)=IBACKI(I,1)+NRS
  160   CONTINUE
C
C                   INCREMENT PERIOD COUNTERS
C
        KEND=1
        ITINV=ITINV+1
C
C                   CREATE NEXT STATISTICS COLLECTION EVENT
C
  200   CONTINUE
       ISTAT = ISTAT + IDSTAT
       CALL ENTER(ISTAT,11,KEND,KEND,KEND)
       RETURN
       END
$       ENDJOB
```

Subroutine: STATUS

Function:

This routine represents a stock status review event (event type 5).

Description:

First, subroutine STATUS calls subroutine STATN (N) for each item N simulated. STATN compares current on hand and on order stocks to the inventory control levels associated with item N. Finally, subroutine STAT enters a new type 5 event on the Future Events List.

STATUS
p. 1 of 1

```
1  04-16-77    12.098
           SUBROUTINE STATUS
           COMMON/NITEM/NITEM
           COMMON/ITIME/ITIME
           COMMON/ISTOCK/ISTOCK,IDSTOC
           COMMON/NDENT/NDENT(50)
           DO 10 N=1,NITEM
           CALL STATN(N)
 1457 DO LOOP INDEX N MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
        10 CONTINUE
           IS=ITIME+IDSTOC
           CALL ENTER(IS,5,0,0,0)
           RETURN
           END
```

Subroutine: STATN

Function:

This routine compares the asset position of item N to its precomputed reorder point, termination, and retention levels. It then initiates buy actions according to the models described in AFLC/ACDRL Working Paper 73, March 1974, by Robert J. Stevens.

Description:

This routine is controlled by the code ICDFOR provided at input through file 05. Eight code values are possible; these are defined in detail in AFLC/ACDRL Working Paper 73.

Basically, each of these alternate computations compare available assets to the reorder level, and initiate ordering actions (using subroutine ORDER) if appropriate. The formulas differ in the types of assets which are added together to compare with the reorder level.

At present, termination and disposal actions are not coded into this routine. Consequently, simulation of termination and disposal policies are not presently possible.

STATN,
p. 1 of 3

```
      SUBROUTINE STATN(N)
*       THIS ROUTINE COMPARES THE ASSET POSITION OF ITEM N
*      TO ITS PRECOMPUTED REORDER POINT,  AND INITIATES BUYS
*      ACCORDING TO THE MODELS ON SERVICEABLE RETURNS DES, IN FOR576
*
       COMMON/IDBUG/IDBUG
       COMMON/ICDFOR/ICDFOR
      COMMON/ITIME/ITIME
      COMMON/ITMNTH/ITMNTH
       COMMON/NBMAX/NBMAX
       COMMON/NLOCBK/NLOCBK
      COMMON/INVACT/INVACT(50)
      COMMON/INVDUE/INVDUE(50)
      COMMON/IROL/IROL(50)
      COMMON/IRQTY/IRQTY(50)
      COMMON/LTADM/LTADM(50)
      COMMON/LTPROD/LTPROD(50)
      COMMON/NBOTU/NBOTU(50)
      COMMON/SRFOR/SRFOR,SRPLT,GDFOR
*
*      COMPUTE INVENTORY POSITION
*
       NRS=NBMAX-NLOCBK
      INPOS=INVACT(N) + INVDUE(N) - NBOTU(N)
       IF(IDBUG.EQ.1)WRITE(6,8013)INVACT(N),INVDUE(N),NBOTU(N),INPOS,NRS
 8013  FORMAT(4X,'****STATN--UNITS OH=',I5,'  DUE-IN=',
&              I5,'  BO=',I5,'  INPOS=',I5,'  REQ-SHORT=',I5)
*
*      GO TO ASSIGNED SERVICEABLE RETURN MODEL
*
      GO TO (91,91,92,92,92,92,91,91),ICDFOR
*
   91 CONTINUE
*
*      MODELS 1,2,7 AND 8 FOR SERVICEABLE RETURNS
*
*      IS INVENTORY POSITION ON OR BELOW REORDER POINT
*
      IEOQ=IRQTY(N)
      INVPOS=INPOS
   99 IF(INVPOS .GT. IROL(N)) RETURN
*
*      COMPUTE TOTAL BUY QUANTITY (EOQ + ROLDF)
*
   97 IBQ=IEOQ + (IROL(N) - INVPOS)
*
*       COMPUTE DELIVERY DATES FOR ORDER
*
   95 IDELIV=ITIME+ITMNTH*(LTADM(N)+LTPROD(N))
*
*       PLACE ORDER FOR IBQ UNITS
*
```

STAT N,
p. 2 of 3

```
      CALL ORDER(N,IBQ,IDELIV)
      RETURN
*
   92 CONTINUE
*
*      COMPUTE AMMENDED INVENTORY POSITION
*
      ISRPLT=SRPLT + .5
      IANPOS=INPOS + ISRPLT
      INVPOS=IANPOS
      IF(ICDFOR .EQ. 3) GO TO 99
      IF(ICDFOR .GE. 5) GO TO 94
*
*      IS AMMENDED INVENTORY POSITION ON OR BELOW REORDER POINT
*      FOR MODEL 4 ONLY
*
      IF(INVPOS .GT. IROL(N)) RETURN
*
       COMPUTE PROCUREMENT CYCLE (EOQ PERIOD)
       AND SERVICEABLE RETURNS EXPECTED IN THIS CYCLE
*
   94 IPC=(FLOAT(IRQTY(N))/GDFOR) + .5
      ISRPC=FLOAT(IPC)*SRFOR + .5
*
*
*      REDUCE REORDER QUANTITY (EOQ) BY EXPECTED SERVICEABLE RETURNS
*      IN PROCUREMENT CYCLE
*
      IEOQ=IRQTY(N) - ISRPC
      IF(ICDFOR .GE. 5) GO TO 98
      IF(IEOQ .LT. 0) IEOQ=0
*
*      COMPUTE BUY QUANTITY FOR MODEL 4
*
      INVPOS=INPOS
      GO TO 97
*
   98 CONTINUE
*
*      IS INVENTORY POSITION ON OR BELOW REORDER POINT FOR MODELS 5 AND6
*
      IF(INPOS .GT. IROL(N)) RETURN
*
*      COMPUTE TOTAL BUY QUANTITY FOR MODEL 5
*
      IBQ=IEOQ + IROL(N) - INPOS
      IF(ICDFOR .EQ. 6) GO TO 96
      IF(IBQ .LE. 0) RETURN
      GO TO 95
*
   96 CONTINUE
*
```

STATN,
p. 3 of 3

```
*       COMPUTE TOTAL BUY QUAMTITY FOR MODEL 6
*
        AEOQ=.5*FLOAT(IRQTY(N))
        IF(FLOAT(IBQ) .LE. AEOQ) IBQ=AEOQ + .5
        IF(FLOAT(IBQ) .GT. AEOQ) IBQ=IBQ
        IF(IBQ .LE. 0) RETURN
        GO TO 95
        END
```

Subroutine: WRIFEL

Function:

This routine writes to File 06 all of the current elements in the Future Events List.

Description:

This routine provides a detailed listing of all information currently recorded in the Future Events List. See Volume I, Chapter IV for a detailed description of routines WRIFEL, INFEL, ENTER, and REMOVE.

02-11-77   10.614

WRIFEL
p. 1 of 1

```
      SUBROUTINE WRIFEL
      COMMON/NFIRST/NFIRST
      COMMON/NENTRY/NENTRY
      COMMON/JPOINT/JPOINT(500)
      COMMON/JESN/JESN(500)
      COMMON/JQTY/JQTY(500)
      COMMON/JPRIOR/JPRIOR(500)
      COMMON/JTYPE/JTYPE(500)
      COMMON/JTIME/JTIME(500)
      COMMON/ILOCFE/ILOCFE(500)
      WRITE(6,8015)JTIME(NFIRST),NFIRST,NENTRY
 8015 FORMAT(///20X,20HEVENTS LIST AT TIME ,3I10//)
      WRITE(6,8020)(K,JTIME(K),JTYPE(K),JESN(K),JQTY(K),JPRIOR(K),JPOINT
     C(K),K=1,NENTRY)
 8020 FORMAT(500(10X,7I10/))
      RETURN
      END
```

Subroutine: TERMIN

Function:

This routine terminates all or part of the most recent replenishment order (if any) still in the administrative phase.

Description:

Subroutine TERMIN terminates all or part of the latest buy (if any) still in the administrative phase (i.e. not yet on contract). The routine continues to terminate orders until either (a) total assets terminated equals the termination quantity or (b) all buys still in the administrative phase have been canceled, whichever occurs first.

By convention, outstanding orders are terminated by setting the order quantity to zero on the Future Events List, and updating the associated pointer structure. The associated receipt event is not removed from the Future Events List, however, until the originally scheduled due-in time.

Note: This routine was not used in the current study, and may not be compatible with other INSSIM subroutines.

TERMIN
p. 1 of 2

```
12-09-76    21.855

      SUBROUTINE TERMIN(KFSN,ITQ)
C        THIS ROUTINE TERMINATES ALL OR PART OF THE LATEST BUY(IF ANY)
C        STILL IN THE ADMINISTRATIVE PHASE (I.E. NOT YET ON CONTRACT)
C        UNTIL EITHER (1)TOTAL ASSETS TERMINATED EQUALS THE TERMINATION
C        QUANTITY (ITQ) OR (2)ALL BUYS STILL IN THE ADMINISTRATIVE PHASE
C        HAVE BEEN CANCELLED, WHICHEVER OCCURS FIRST.
      COMMON/IDBUG/IDBUG
      COMMON/IOBLIG/IOBLIG
      COMMON/ITIME/ITIME
      COMMON/ITMNTH/ITMNTH
      COMMON/INVDUE/INVDUE(50)
      COMMON/LTPROD/LTPROD(50)
      COMMON/NORDPT/NORDPT(50)
      COMMON/UCOST/UCOST(50)
      COMMON/JQTY/JQTY(500)
      COMMON/JTIME/JTIME(500)
      COMMON/JPRIOR/JPRIOR(500)
      COMMON/ITERM /ITERM(30,3)
      COMMON/INVOR/INVOR(30,3)
      IF(IDBUG.NE.1) GO TO 18
      WRITE(6,8000)ITIME,KFSN,ITQ,NORDPT(KFSN)
 8000 FORMAT(7H TERMIN,I10,8I8)
   18 CONTINUE
C        ARE THERE ANY ORDERS OUTSTANDING ON ITEM KFSN
      IF(NORDPT(KFSN).LE.0) RETURN
C        OBTAIN LOCATION OF ORDER DATA ON FUTURE EVENTS LIST
   20 KORD=NORDPT(KFSN)
C        COMPUTE THE DATE THAT THE MOST RECENT PURCHASE ORDER CLEARS
C        THE ADMINISTRATIVE PHASE.
   25 IDUE=JTIME(KORD)
      IADM=IDUE-ITMNTH*LTPROD(KFSN)
C        IS THE ORDER ALREADY ON CONTRACT
      IF(IADM.LT.ITIME) RETURN
C        DOES THE ORDER QUANT. EXCEED THE TERMINATION QUANTITY.
      IF(JQTY(KORD).LE.ITQ) GO TO 40
C        DECREASE THE ORDER BY THE TERMINATION QUANTITY.
      JQTY(KORD)=JQTY(KORD)-ITQ
C        UPDATE ON-ORDER STATUS RECORDS
      INVDUE(KFSN)=INVDUE(KFSN)-ITQ
      IOBLIG=FLOAT(IOBLIG)-FLOAT(ITQ)*UCOST(KFSN)
      JTQ=-ITQ
      CALL CUM(ITERM,ITQ,KFSN)
      CALL CUM(INVOR,JTQ,KFSN)
      RETURN
C   *** NOTE ***
C          BY CONVENTION,OUTSTANDING ORDERS ARE TERMINATED BY SETTING
C          THE ORDER QUANTITY TO ZERO ONTHE FUTURE EVENTS LIST,
C          AND UPDATING THE FSN CROSS REFERENCE STRUCTURE.  THE RECEIPT
C          TRANSACTION IS NOT REMOVED FROM THE FUTURE EVENTS LIST HOWEVER
C          UNTIL THE ORIGINALLY ANTICIPATED DUE-IN-TIME.
C
C        UPDATE ON-ORDER STATUS RECORDS
```

TERMIN
p. 2 of 2

```
      40 KQTY=JQTY(KORD)
         JQTY(KORD)=0
         NORDPT(KFSN)=JPRIOR(KORD)
         INVDUE(KFSN)=INVDUE(KFSN)-KQTY
         IOBLIG=FLOAT(IOBLIG)-FLOAT(ITQ)*UCOST(KFSN)
         JTQ=-KQTY
         CALL CUM(ITERM,KQTY,KFSN)
         CALL CUM(INVOR,JTQ,KFSN)
         ITQ=ITQ-KQTY
C          ARE THERE ANY OTHER ORDERS OUTSTANDING ON ITEM KFSN
         IF(NORDPT(KFSN).GT.0) GO TO 20
         RETURN
         END
```

Subroutine: ZERO

Function:

This routine zeros the INSSIM statistical accumulators.

Description:

This routine is called at the beginning of each INSSIM run to zero the variables for recording performance statistics, beginning on hand and on order stocks, and obligations.

ZERO

p. 1 of 2

```
1Ø     $     IDENT   WPØ287,ADDSC-CALLAHAN
2Ø     $     FORTY   DECK
3Ø     $     PRMFL   C*,W,S,REQS/ZERO.O
4Ø     $     REMOTE  P*,AC
5Ø     $     REMOTE  $$,AC
6Ø          SUBROUTINE ZERO
7Ø          COMMON/IOBLIG/IOBLIG
8Ø          COMMON/IBOP/IBOPOH(3),IBOPOR(3)
9Ø          COMMON/IBAUTH/IBAUTH(12)
1ØØ         COMMON/IBPROJ/IBPROJ(12)
11Ø         COMMON/IBACKI/IBACKI(3Ø,3)
12Ø         COMMON/IBACKT/IBACKT(3Ø,3)
13Ø         COMMON/IBAKDI/IBAKDI(3Ø,3)
14Ø         COMMON/IBAKDT/IBAKDT(3Ø,3)
15Ø         COMMON/ICANCL/ICANCL(3Ø,3)
16Ø         COMMON/IDISPS/IDISPS(3Ø,3)
17Ø         COMMON/IEXPED/IEXPED(3Ø,3)
18Ø         COMMON/IFILLI/IFILLI(3Ø,3)
19Ø         COMMON/IFILLT/IFILLT(3Ø,3)
2ØØ         COMMON/INVDAY/INVDAY(3Ø,3)
21Ø         COMMON/INVOH/INVOH(3Ø,3)
22Ø         COMMON/INVOR/INVOR(3Ø,3)
23Ø         COMMON/IORDER/IORDER(3Ø,3)
24Ø         COMMON/IRATON/IRATON(3Ø,3)
25Ø         COMMON/IRECET/IRECET(3Ø,3)
26Ø         COMMON/IREQC /IREQC(3Ø,3)
27Ø         COMMON/IREQI /IREQI(3Ø,3)
28Ø         COMMON/IREQT/IREQT(3Ø,3)
29Ø         COMMON/IRETRN/IRETRN(3Ø,3)
3ØØ         COMMON/ISHIPI/ISHIPI(3Ø,3)
31Ø         COMMON/ISHIPT/ISHIPT(3Ø,3)
32Ø         COMMON/ITERM /ITERM(3Ø,3)
Ø325         COMMON/ISMORD/ISMORD(3Ø,3)
Ø326         COMMON/ILGORD/ILGORD(3Ø,3)
Ø327         COMMON/IBOPCT/IBOPSM(3),IBOPLG(3)
33Ø    C       ZERO STATISTICAL ACCUMULATORS
34Ø         IOBLIG=Ø
35Ø         DO 25 I=1,3Ø
36Ø         DO 25 J=1,3
37Ø         ICANCL(I,J)=Ø
38Ø         INVOH(I,J)=Ø
39Ø         INVOR(I,J)=Ø
4ØØ         IRECET(I,J)=Ø
41Ø         IRETRN(I,J)=Ø
42Ø         INVDAY(I,J)=Ø
43Ø         IORDER(I,J)=Ø
```

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 77-1235	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE INVENTORY SYSTEM SIMULATOR (INSSIM), VOLUME II: PROGRAM LISTINGS AND NARRATIVES		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER RM 77-03
7. AUTHOR(s) W. Steven Demmy		8. CONTRACT OR GRANT NUMBER(s) F49620-77-C-0063 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Systems 3575 Charlene Drive Dayton, OH 45432		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A5
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, DC 20332		12. REPORT DATE August 1977
		13. NUMBER OF PAGES 116
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

single-location, multi-item, EOQ, inventory, simulation model, FORTRAN

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the features of the Inventory System Simulator (INSSIM), a FORTRAN-based simulation model of a general single-location inventory system. The model was specifically designed to evaluate alternate forecasting and inventory management policies proposed for use in the EOQ Buy Computation System (D062). Volume I of this report describes the philosophy, organization, and input/output features of INSSIM. Volume II documents the FORTRAN statement listings of each INSSIM program, and provides narratives describing the functions of each routine.

```
      IDISPS(I,J)=0
      ITERM(I,J)=0
      IEXPED(I,J)=0
      IRATON(I,J)=0
      IREQC(I,J)=0
      IREQT(I,J)=0
      IREQI(I,J)=0
      IBACKT(I,J)=0
      IBACKI(I,J)=0
      IBAKDT(I,J)=0
      IBAKDI(I,J)=0
      IFILLT(I,J)=0
      IFILLI(I,J)=0
      ISHIPT(I,J)=0
      ISHIPI(I,J)=0
      ISMORD(I,J)=0
      ILGORD(I,J)=0
   25 CONTINUE
C
C        ZERO ON-HAND AND ON-ORDER COUNTERS
C
      DO 110 I=1,3
      IBOPOH(I)=0
      IBOPOR(I)=0
      IBOPSM(I)=0
      IBOPLG(I)=0
  110 CONTINUE
      RETURN
      END
$       ENDJOB
```